

WIN AN ATARI TOUCH TABLET!!

ATARI USERS MAGAZINE........AN ATARI USERS MAGAZINE........AN ATARI USERS M

......GAMES........ATARI........NEWS........ATARI........PROGRAMS........ATARI........REVIEWS......

Llamasoft

ORIGINAL SOFTWARE DESIGN

49 MOUNT PLEASANT, TADLEY, HANTS. RG26 6BN.

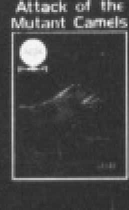
SEND S.A.E. FOR (FREE NEWS LETTER) "THE NATURE OF THE BEAST".












TRAXX	HELL GATE	METAGALACTIC LLAMAS BATTLE	ABDUCTOR	MATRIX	LASERZONE	GRIDRUNNER	SHEEP IN SPACE	HOVER BOVVER	ATTACK OF THE MUTANT CAMELS	REVENGE OF MUTANT CAMELS
VIC 20 LLA 12003 £5.00	VIC 20 LLA 12007 £6.00 CBM 64 LLA 21008 £5.00	VIC 20 LLA 11006 £5.50 CBM 64 LLA 21010 £5.50	VIC 20 LLA 11001 £5.00	VIC 20 LLA 12005 £5.00 CBM 64 LLA 21006 £6.00	VIC 20 LLA 12004 £5.00 CBM 64 LLA 21004 £6.00	VIC 20 LLA 11002 £5.00 CBM 64 LLA 21003 £5.00 ATARI LLA 41001 £5.00	CBM 64 LLA 21009 £7.50	CBM 64 LLA 21005 £7.50	CBM 64 LLA 21001 £6.00 ATARI LLA 41003 £7.50	CBM 64 LLA 21007 £7.50


WOOLWORTH W H SMITH

John Menzies


DEALERS may order direct from CBS Telesales 01-960 2155 quoting account number, LLA code number and quantities. Goods delivered within 48 hours.



Editor & Publisher
Les Ellingham
Printed by
Birbeck & Sons Ltd.
Typesetting
Budget Typesetting Ltd.
from copy set on an Atari 800 and transmitted by modem.
Editorial & Advertising
0785 41153
Correspondence
PAGE 6 Magazine
P.O.Box 54
Stafford
ST16 1DR

PAGE 6 is published bi-monthly

Page 6 is a users magazine and relies entirely on readers' support in submitting articles and programs. The aim is to explore Atari computing through the exchange of information and knowledge and whilst we cannot, unfortunately, pay for articles published, we hope that you will gain satisfaction from seeing your work published and in turn we hope that you will learn from articles submitted by other readers.

ISSUE 11 **September/October 1984**

Subscription rates - annual (6 issues)

U.K.	£7.00	Outside Europe - Surface	£10.50
Europe	£10.50	Outside Europe - Airmail	£16.00

Single copies and back issues at one-sixth of above rates.

Please make cheques payable to PAGE 6.

Please mention PAGE 6 when replying to advertisers.

Editorial

A NEW DAWN?

This issue's Editorial was to have been a criticism of Atari's continuing lack of support for their computers prompted by a press release which started "Long live the Video game!" but of course everything has now changed. What was Atari policy is no longer Atari policy. By the time you read this, Atari's new path will have been charted and let us all hope that this time they are heading in the right direction.

For those of you who do not read the U.K. micro magazines, all of this has to do with the take-over of Atari by Jack Tramiel, founder and ex-President of Commodore. In fact, even if you do read the micro magazines, you may not have learned much as the take-over was covered much more extensively in the Financial Times, but then the U.K. computer press has never taken much interest in Atari. Maybe now that will all change.

Many Atari owners seem to have taken events as the toll of doom for Atari, as the throwing away of a great computer to the opposition, but I think that Atari owners may now be well and truly on the verge of a new dawn. A dawn that will finally bring true recognition of the fact that Atari has always produced the finest computer on the market. I hope that in saying this I don't follow Compute!'s unfortunate classic of bad timing when they published, in their July issue, an interview with James Morgan on the future of Atari. By the issue date, James Morgan was no longer CEO of Atari and the whole structure of the company had changed! Such are the perils of writing copy in advance. I have no doubt that much more will develop between my writing this and publication.

Every reader who has stayed loyal to Atari computers for any length of time must have realised that Atari was slowly sinking into obscurity, still riding the VCS wave, still failing to realise what a great computer system they had and still misunderstanding the U.K. market. Despite all the promise nothing had really changed by mid-summer and with plans to launch a new video game system in the autumn, which would compete with their own computers, the future for the computer side of Atari in the U.K. looked even bleaker. Warner Bros had been talking to Phillips for some time but in the end they made what was perhaps one of their wisest decisions since buying Atari, they sold the company to the man that more than any other, outside Atari, put Atari in the sorry position it was in.

Jack Tramiel founded and built Commodore and he put Commodore at the forefront of home computing. Not only did he put Atari in the shadows, he completely eclipsed them and not once did he mention a video games machine. He was offered the arcade machine side of Atari but turned it down and has been quoted as saying that the video game machine is dead, which leaves us with a company that we have all wanted all along - a company totally dedicated to the home computer. At the time of writing, reports from the U.S. state that the workforce had been reduced to 200 and the price of the 800XL dropped from $250 to $150 to bring it in direct competition with Commodore. U.K. prices were expected to drop to £199.99 for the 800XL and £99.99 for the 600XL. Mr Tramiel is said to have plans to drop the VCS entirely 'within six months' and to introduce a computer to compete with the Apple II. He is also said to be considering, for next year, a direct competitor for Apple's Mackintosh. All of which seems to be the direction in which we would all like Atari to go.

Atari have always had superb products but they have in the past lacked understanding of both their own products and marketing in general. Jack Tramiel has proved with Commodore that he fully understands the marketing of home computers and he now has the opportunity to marry one of the best marketing strategies in the business with the best products. It won't be so easy this time round because Commodore won't make it as easy for Atari as Atari made it for Commodore but it promises to be the most exciting time since the early days when Atari took the home computer world by storm.

If Atari had stayed with Warner Bros, those of us who have remained loyal would probably have witnessed the end of a dream. It may still happen but I think not. Atari is dead, long live Atari!

On a personal note, please be patient if you find any delays between now and Issue 12 on anything ordered or if the next one is a few days late. A new Atari fanatic is due to be born on 27th September - right in the middle of the typesetting preparation for the next issue!

Les Ellingham

Editor

News and New Products

Over the summer the micro-world hibernates. You stop buying software and software producers have so few new products. Spectrum programmers jet off to the Bahamas whilst Atari programmers have to take holiday jobs to survive another year. All of which means that news and new products are thin on the ground.

Hottest news of a hot July is of course the take-over of Atari but it is all happening too close to copy date to report anything specific. Expect big price reductions on hardware and software and the 800XL as the only machine to survive into the autumn.

English Software has **STRANDED** out, a 35 screen graphic adventure in 32K. **ATTACK OF THE MUTANT CAMELS** from Llamasoft brings (hopefully) a new range of top class arcade games at amazing prices (see review).

In the States (info courtesy of The Pokey Press), MUSE SOFTWARE should be releasing **BEYOND CASTLE WOLFENSTEIN** and PARKER BROS should have **FROGGER II: THREEDEEP** and **MONTEZUMA'S REVENGE**, an action adventure featuring 'Panama Joe' and a 100 room maze to investigate. Also **STAR WARS: THE ARCADE GAME** and **GYRUSS**.

A couple of items received for review recently deserve mention. Firstly a **RUBBER KEYBOARD** from **FILESIXTY** which looks to be a very handy piece of equipment for 400 owners. Secondly **AUTOTECT** from **Magical Electronic Services** which is a write-protect/enable switch to attach to the 810 requiring no soldering. Looks excellent. Both of these arrived too late for reviews in this issue but full reports will follow.

Rumours from England .... a disk drive at under £200 made over here to an American design .... a reliable independent recorder at under £30 .... a cassette interface allowing a normal stereo recorder to be used .... an interface for a modem .... and, finally, an advertising campaign from Atari (please!)

Since the magazine increased in size a couple of issues ago the postage costs have increased considerably. We have tried to absorb these costs but with U.K. postage due to increase in September we are forced to increase the subscription price. If you work it out we still don't pass on the full cost of post and packing and we hope that we continue to receive you subscription support. You still get six issues of PAGE 6 for less (in most cases) than one piece of software.

Readers Letters

AN EDITOR REPLIES

Dear PAGE 6,

In issue 8, Mr B. of Herts encourages Atari owners to write to magazines and software houses asking for more and better support. "Your voice does count", he exclaims. He is right. Editors have only a hazy idea what their readers want, but are in the business of trying to supply it. All feedback makes an impact.

In my own magazine, Practical Computing, we regularly publish a lot of material for the Research Machines RML 380Z. This is because lots of teachers read Practical Computing, they send in lots of well written material and we publish it. It is not because the 380Z is an enormous seller.

As an Atari owner myself, I want Practical Computing to cover the Atari as well as possible. This includes fairly regular games reviews and an "Open File" program section, but (a) we very rarely get sent programs for review, except by Atari and English Software - we never get sent the expensive American imports. Also (b) the number of short, usable readers programs sent in is fairly small. The Sharp and Research Machines micros do better, as well as all the obvious ones!

Magazines usually beg, borrow or buy the machines they need to run software, we have over a dozen. Some manufacturers help with long term loans and Commodore, for example, sells 64's to journalists at half price. When the staff have machines, software and readers programs tc run, they tend to get to know those micros better than others and the bandwagon starts to roll....

Don't blame the magazines for their ignorance of Atari. Follow Mr B's advice, get out your keyboards or pens and start writing!

Jack Schofield
Editor
Practical Computing

THE COMPUTER JUNGLE?

Dear PAGE 6,

Computer terminology is baffling enough but for Atari owners it is getting worse.

Imagine hearing someone say "I have my Gorilla Banana tied to an Ape-Face and use Elephants in my Rana". Maybe Jungle Book should be required reading instead of the Basic Manual!

Alan Hollis
BFPO 40

BLEEP!

Dear Sirs,

Here is a tip that newer owners may not know.

When loading or saving a program press CTRL and 2 together and when the cassette has finished the computer will bleep at you to let you know that it has finished. Much better than sitting watching an empty screen.

Kevin Ramshaw
Tyne & Wear

GOING DUTCH

Dear Les,

With the introduction of the Atari home computers in the Dutch market, the moment has arrived to found an Atari Users Group and the initiative has been taken by three 'founding fathers' who are members of HCC, the large Dutch Hobby Computer Club.

The User Group is now being constituted within the framework of HCC and Dutch and Flemish readers of this journal are invited to get in touch with Nic Oosterbaan, Raadhuislaan 114, Voorschoten, The Netherlands, phone 01717-2555. Also there is a HCC-Belgium and we would welcome hearing from French-speaking members of that organisation as well as anyone interested in our work. At the time of writing we have about thirty members half of whom are more or less able to read English and several who read and write French.

Authors of published and unpublished programs are invited to send us their material for inclusion in our library and possible Dutch translation. Apart from our recognition we shall endeavour to repay you in kind with disks or cassettes of suitable material.

Nic Oosterbaan
Holland



Listing Conventions

The program listings in PAGE 6 are prepared so that the listings match exactly what you see on a normal 38 column screen. Inverse video and CONTROL characters appear as they do on the screen.

To obtain CTRL characters use the keys shown in the following chart.

Normal Video	TYPE THIS	Inverse Video
♥	CTRL ,	
	CTRL A	
	CTRL B	
	CTRL C	
	CTRL D	
	CTRL E	
	CTRL F	
	CTRL G	
	CTRL H	
	CTRL I	
	CTRL J	
	CTRL K	
	CTRL L	
	CTRL M	
	CTRL N	
	CTRL O	
♣	CTRL P	
	CTRL Q	
	CTRL R	
	CTRL S	
	CTRL T	
	CTRL U	
	CTRL V	
	CTRL W	
	CTRL X	
	CTRL Y	
	CTRL Z	
♦	CTRL .	
♠	CTRL ;	
	SHIFT =	
	ESC ESC	
↑	ESC CTRL -	
↓	ESC CTRL =	
←	ESC CTRL +	
→	ESC CTRL *	
	ESC SHIFT CLEAR	
	ESC DELETE	
	ESC TAB	
	ESC SHIFT DELETE	
	ESC SHIFT INSERT	
	ESC CTRL TAB	
	ESC SHIFT TAB	
	ESC CTRL 2	
	ESC CTRL DELETE	
	ESC CTRL INSERT	

Make sure that you SAVE a copy of any listing before you attempt to RUN it.

Review

BARGAIN TIME

THREE FROM P.F.SOFTWARE

In last issue's editorial I mentioned that there were a number of low price programs around which quite frankly put you off from responding to advertisements but I also said that there were some excellent programs available. The problem is how do you know which is which?

With the above in mind I decided to take a look at three programs from P.F.Software which range in price from £2.50 to £4.50 and you won't get much cheaper than that! Most owners do not realise that Atari software generally sells only in small quantities and the cost of fancy packaging and advertising often means that selling software at low prices is not feasible. Forget the thoughts of the idle rich Atari programmer, it is simply not true! In order to sell at such low prices, P.F. Software have cut right back on the packaging to the extent that you get a typed label and photocopied instructions. You may be dismayed initially at having spent even £2.50 but it is the programs themselves that count, so are they worth it?

The first of the trio at £2.50 is **Blackjack** which is the standard game of pontoon using the joystick to twist or stick and play against the dealer. A very familiar game that has almost become a computer standard by now but this version uses excellent high-resolution graphics in four colours and knocks Atari's own Blackjack for six. Especially impressive are the court cards with good design and plenty of colour. It really does look like a pack of cards on the screen. Against the £9.95, or whatever, that Atari charge for a very basic program this is undoubtedly worth every penny of £2.50!

Secondly, for those of you who are budding artists but do not have a disk drive or cannot afford MicroPainter or Paint comes **Art Atari** which is a drawing utility enabling you to create screen pictures and save them to cassette. It obviously lacks the sophistication of the MicroPainter type utility but it does allow you to compose pictures in up to 80 colours and has the usual line, draw and fill functions. The different colours are achieved by using variable display list interrupts and although there are some limitations on how the colours are used, with careful planning some superb hi-res pictures can be composed. Generally you will need only a dozen or so colours in any given drawing and the results that can be obtained are very impressive. Included on the tape is a demo picture which is copied from an original drawn with MicroPainter and whilst Art Atari is not so easy to use, the end result is just as impressive. If you are looking for a drawing utility but are not sure whether you can get on with one, Art Atari will get you started for very little cost. You can always go on to MicroPainter afterwards. Far better than spending £30 and finding that you don't like drawing programs after all!

To my mind, the best of the three is **Picture Puzzle**. I was amazed at the quality and the program will give you many hours of enjoyment if you like the 'sliding square' type of puzzle. If you have young children then the program will be of extra value for the easier levels are ideally suited to young minds. Picture Puzzle is very similar to the range put out by Thorn EMI some time ago and consists of a high resolution picture which is scrambled up and then needs to be re-arranged to the original using the joystick. There are five difficulty levels and each picture can be divided into 16, 20, 25 or 40 pieces. As the difficulty level increases the pieces are more jumbled and on level 5 the screen is blanked while the pieces are moved. Choose this level and forty pieces and you could be in for a long night! At the opposite end on level one and using only 16 pieces, the program is ideally suited to young children who, maybe with a little help, can easily re-arrange the picture and will gain a lot of pleasure in putting it back together again. There are two pictures to choose from and the quality of both and of the program in general is very good. If you like picture puzzles you will probably consider your £3.50 well spent.

P.F. Software is obviously a 'home base' company putting out some well written software at pocket money prices. You don't get fancy packaging and fancy protection techniques and these are not of the top American (and now British) standards but a great deal of care has gone into making the programs pesentable, and you are not being asked to pay fancy prices.

Blackjack and Art Atari come on 16K cassette and Picture Puzzle requires 32K. ●

48K Spectrum & Atari 400/600/800



Blue Thunder

by Richard Wilcox

After skilfully piloting your Jetcopter through a bombardment of Electronic Storms, Ground & Sea Based Missiles and attack from Armoured Barrage Balloons, you thought you were safe to complete your mission ... But No! ... yet another wave of deadly Jet Fighters appear from nowhere with only one objective...to eliminate you!

Only your Hyper-Phase Lasers and amazing arcade skill can keep you alive to complete a task that looks impossible. As the only survivor of a once mighty invasion force you must penetrate deep beyond enemy lines. From a remote island protected by a complex defensive screen you must rescue your wounded comrades held captive beside an unstable nuclear reactor, which you must first destroy!

The ultimate in 100% Machine Code Arcade Games from a Master Programmer, achieving a new peak in programming perfection and super smooth movement. Blue Thunder sets astounding new standards in Hi-Res Graphics with Pixel Scrolling over 6 screens 5 different missions, Hi-Score and incredible breathtaking PERSPECTIVE GRAPHICS!

Richard Wilcox Software carries a lifetime guarantee, should any of our games fail to load please return them for an immediate replacement.

All Richard Wilcox Software is available from leading software outlets or by completing the attached coupon.

Richard Wilcox Software

STATION ROAD, WALSALL, WS7 0JZ

To receive your advance copy of BLUE THUNDER before it goes on general release complete and post this coupon to us today! We will rush you your copy by return of post!

☐ 48K Spectrum (Cassette Only) £ 5.50

☐ Any Atari 400/600/800 ☐ **Commodore 64 £6.95**

☐ Cassette £ 9.95 ☐ Disk £ 11.95

Please rush me ..... copy/ies by return as ticked above.

I enclose a Cheque/Postal Order made payable to:

RICHARD WILCOX SOFTWARE for £..............

Name..

Address..

....

Telephone......................................

SEND TO: RICHARD WILCOX SOFTWARE
Station Road, Walsall, WS7 0JZ

TRADE ONLY ENQUIRIES WELCOME ON 05433 2787

GRID

by Mark Hewson

REQUIRES 32K

Grid is a two player game requiring the use of two joysticks. After a lengthy initialisation (approx. 50 seconds) a grid appears and the players are offered a choice of four game options. The desired option is selected by pressing the trigger button as the option lights up.

The object of the game is to outscore your opponent by placing your coloured pieces strategically to either gain high scores yourself or to stop your opponent from obtaining a high score.

Scoring is carried out in four directions.

1. Top to bottom
2. Top left to bottom right
3. Bottom left to top right
4. Left to right

The scoring proceedure is difficult to explain but the following examples should help.

Top to bottom	0
Top left to bottom right	0
Bottom left to top right	0
Left to right	2
Total	2

Top to bottom	2
Top left to bottom right	0
Bottom left to top right	0
Left to right	2
Total	4

Top to bottom	2
Top left to bottom right	2
Bottom left to top right	0
Left to right	2
Total	6

Top to bottom	2
Top left to bottom right	2
Bottom left to top right	3
Left to right	3
Total	10

No score is allowed for one piece on its own in any direction. The best way to become familiar with the scoring routine is to play a few games and study the grid carefully.

There are several "Mystery Squares" and if you place your piece on these there is an even chance of either doubling your score or scoring no points for that move. In the "Tricky" version of the game Mystery Squares have an additional function. If you choose a 'no score' square not only do you score zero for the move but your opponents piece will be placed on the square."Tricky" games have red question marks over the Mystery squares and Normal games have green question marks.

I am indebted to ANTIC magazine - Volume 2 Issue 3 - for the machine code Player Missile routine.

```
10 REM ******************************
11 REM *           GRID             *
12 REM * A two player strategy game *
13 REM *            by              *
14 REM *        MARK HEWSON         *
15 REM ******************************
16 REM
100 CLR :POKE 832,6
110 DIM TOTAL(2),P$(4),S(9),E$(10):GO=
1:P$="[illegible]"
120 S(1)=0:FOR I=2 TO 8:S(I)=I:NEXT I
130 GOSUB 2160
140 GOSUB 1080:POKE 54286,192
150 POKE 709,0:GOSUB 1940
160 TOTAL(1)=0:TOTAL(2)=0
170 GOSUB 1830
180 REM CURSOR
190 PA=96:PD=87:POKE 1536,0:POKE 704,1
4
200 RESTORE 220:FOR I=1571 TO 1578:REA
D X:POKE I,X:NEXT I
210 POKE 1611,7:POKE 53256,0
220 DATA 0,24,24,126,126,24,24,0
230 IF MOVES=0 THEN GOTO 610
240 POKE 53248,PA:POKE 1536,PD:FOR DE=
1 TO 50:NEXT DE
250 A=STICK(GO-1)
260 IF STRIG(GO-1)=0 THEN POKE 77,0:GO
TO 350
270 Z=PEEK(53279)
280 IF Z=6 THEN POKE 559,0:FOR X=53248
 TO 53251:POKE X,0:NEXT X:MOVES=0:? CH
R$(125):GOTO 150
290 IF A=15 THEN GOTO 250
300 IF A=13 AND PD<80 THEN PD=PD+8:GOT
O 240
310 IF A=14 AND PD>40 THEN PD=PD-8:GOT
O 240
```

```
320 IF A=11 AND PA>96 THEN PA=PA-8:GOT
0 240
330 IF A=7 AND PA<152 THEN PA=PA+8:GOT
0 240
340 GOTO 250
350 REM TRIG PRESSED
360 BONUS=1:A=(PA-48)/8:D=(PD-23)/8:LO
CATE A,D,Z
370 IF Z=91 THEN POSITION A,D:? #6;P$(
GO,GO):SOUND 0,10,10,14:SOUND 1,15,10,
14:GOTO 420
380 M=INT(RND(0)*2)
390 IF M=1 AND (Z=220 OR Z=203) THEN 1
660
400 IF M=0 AND (Z=220 OR Z=203) THEN 1
490
410 POKE 704,0:FOR Q=220 TO 255 STEP 0
.5:SOUND 0,Q,10,10:NEXT Q:SOUND 0,0,0,
0:POKE 704,14:GOTO 250
420 REM SCORE
430 N=0:S=0:A1=A:D1=D
440 LOCATE A1,D1,Z:IF Z=ASC(P$(GO,GO))
 THEN D1=D1-1:GOTO 440
450 LOCATE A1,D1+1,Z:IF Z=ASC(P$(GO,GO
)) THEN D1=D1+1:N=N+1:GOTO 450
460 S=S+S(N)*BONUS
470 N=0:A1=A:D1=D
480 LOCATE A1,D1,Z:IF Z=ASC(P$(GO,GO))
 THEN D1=D1-1:A1=A1-1:GOTO 480
490 LOCATE A1+1,D1+1,Z:IF Z=ASC(P$(GO,
GO)) THEN D1=D1+1:A1=A1+1:N=N+1:GOTO 4
90
500 S=S+S(N)*BONUS
510 SOUND 0,0,0,0:SOUND 1,0,0,0
520 N=0:A1=A:D1=D
530 LOCATE A1,D1,Z:IF Z=ASC(P$(GO,GO))
 THEN A1=A1-1:GOTO 530
540 LOCATE A1+1,D1,Z:IF Z=ASC(P$(GO,GO
)) THEN A1=A1+1:N=N+1:GOTO 540
550 S=S+S(N)*BONUS
560 N=0:A1=A:D1=D
570 LOCATE A1,D1,Z:IF Z=ASC(P$(GO,GO))
 THEN A1=A1-1:D1=D1+1:GOTO 570
580 LOCATE A1+1,D1-1,Z:IF Z=ASC(P$(GO,
GO)) THEN A1=A1+1:D1=D1-1:N=N+1:GOTO 5
80
590 S=S+S(N)*BONUS
600 TOTAL(GO)=TOTAL(GO)+S
610 GO=3-GO:AA=170:IF GO=1 THEN AA=78
620 POKE 656,0
630 POKE 657,6:? "VNTUO";
640 POKE 657,29:? "VNTUO";
650 POKE 656,1
660 POKE 657,8-INT(LEN(STR$(TOTAL(1)))
/2):? TOTAL(1);
670 POKE 657,31-INT(LEN(STR$(TOTAL(2))
)/2):? TOTAL(2);
680 IF MOVES=56 THEN POKE 53249,0:GOTO
 1190
690 POKE 53249,AA:POKE 53248,PA
700 IF MOVES=0 THEN POKE 1536,PD:FOR X
=10 TO 0 STEP -1:FOR I=100 TO 50 STEP
-4:SOUND 0,I,10,X:NEXT I:NEXT X
710 MOVES=MOVES+1:GOTO 250
720 REM TUNES
730 Q=12:D=10
740 IF RE=0 THEN RESTORE 840
750 IF RE=1 THEN RESTORE 860
760 D=D+16:READ S0,S1,S2,DE:POKE 540,D
E*Q
770 IF S0=999 THEN SOUND 0,0,0,0:SOUND
 1,0,0,0:SOUND 2,0,0,0:POKE 709,202:GO
TO 2070
780 SOUND 0,S0,14,10:SOUND 1,S1,10,10:
SOUND 2,S2,10,10
790 IF PEEK(540)=0 THEN 760
800 Z=PEEK(53279)
810 IF Z=6 THEN SOUND 0,0,0,0:SOUND 1,
0,0,0:SOUND 2,0,0,0:POKE 709,202:GOTO
280
820 D=D+16:IF D>255 THEN D=10
830 POKE 709,D:GOTO 790
840 DATA 0,40,50,2,40,50,60,2,50,0,40,
2,100,200,0,4,0,0,0,1,50,60,70,2,0,50,
100,2,130,230,50,2,200,120,30,4
850 DATA 999,999,999,0
860 DATA 150,120,0,2,100,80,0,2,90,75,
0,2,165,110,0,2,150,90,0,2,200,150,0,1
.75,0,0,0,0,200,150,120,4
870 DATA 999,999,999,0
880 REM CHOICES
890 IF STRIG(0)=0 OR STRIG(1)=0 THEN 8
90
900 POKE 656,0:POKE 657,1:POKE 755,2:?
 "six     ?  twelve ?  six     !  twelve
 !";
910 POKE 656,1:POKE 657,1:? "normal @
 normal @  tricky @  tricky @";
920 POKE 1538,103:POKE 709,202
930 POKE 53250,52:POKE 706,144:SOUND 0
,125,10,4:MM=6:RE=0:GOSUB 970
940 POKE 53250,92:SOUND 0,100,10,4:MM=
12:RE=0:GOSUB 970
950 POKE 53250,132:POKE 706,80:SOUND 0
,75,10,4:MM=6:RE=1:GOSUB 970
960 POKE 53250,172:SOUND 0,50,10,4:MM=
12:RE=1:GOSUB 970:GOTO 930
970 FOR I=1 TO 3
980 POKE 656,3:POKE 657,12:? "
 ";
990 FOR DE=1 TO 15:IF STRIG(0)=0 OR ST
RIG(1)=0 THEN 1060
1000 NEXT DE
1010 POKE 656,3:POKE 657,12:? "trigger
 selects";
1020 FOR DE=1 TO 15:IF STRIG(0)=0 OR S
TRIG(1)=0 THEN 1060
1030 NEXT DE
1040 NEXT I
1050 RETURN
1060 ? CHR$(125):POKE 53250,0:POKE 706
,196:POKE 1538,131:POKE 709,202:POP :P
OP :POP
1070 SOUND 0,0,0,0:POKE 77,0:FOR DE=1
TO 80:NEXT DE:GOTO 2060
1080 REM BARS
1090 POKE 623,1:POKE 1613,7:POKE 1614,
7:POKE 53258,3:POKE 53259,3:POKE 53250
,0:POKE 53251,0
1100 POKE 706,0:POKE 707,52:POKE 1538,
103:POKE 1539,103
1110 FOR I=1591 TO 1598:POKE I,255:NEX
T I
1120 FOR I=1601 TO 1608:POKE I,255:NEX
T I
1130 REM ARROW
1140 POKE 705,119:RESTORE 1170
1150 FOR I=1581 TO 1588:READ X:POKE I,
X:NEXT I
1160 POKE 1612,7:POKE 1537,92
1170 DATA 60,60,60,60,255,126,60,24
1180 AA=78:RETURN
1190 REM FINISH
1200 Z=10:E$="   NUIX   ":C=136
1210 IF TOTAL(2)>TOTAL(1) THEN E$=" UC
N XOSL ":C=72:GO=1
1220 IF TOTAL(1)>TOTAL(2) THEN E$="PLC
CS XOSL":C=202:GO=2
1230 POKE 710,C:POKE 755,1
1240 POSITION 5,9:? #6;E$
1250 FOR I=129 TO 13 STEP -2:SOUND 0,I
,10,12:SOUND 1,I-4,10,10:SOUND 2,I-7,1
0,10:SOUND 3,I-12,10,10:NEXT I
1260 FOR I=0 TO 3:SOUND I,0,0,0:NEXT I
1270 GOSUB 1420:GOSUB 1420
1280 D=5:FOR I=1 TO 3
1290 SOUND 0,100,10,Z:GOSUB 1420
1300 SOUND 0,75,10,Z:GOSUB 1420
1310 SOUND 0,50,10,Z:GOSUB 1420
1320 SOUND 0,75,10,Z:GOSUB 1420
1330 NEXT I
1340 SOUND 0,100,10,Z:GOSUB 1420:GOSUB
 1420:SOUND 0,0,0,0
1350 POKE 656,2:POKE 657,15:X=0:D=6
1360 ? "YRIZ IPIOS";
1370 X=X+1:POKE 755,0:GOSUB 1420
1380 POKE 755,1:GOSUB 1420
1390 IF X<20 THEN 1370
1400 Z=Z-2:IF Z<0 THEN Z=0
1410 GOTO 1280
1420 DE=0
1430 DE=DE+1:IF DE=D THEN RETURN
1440 C=C+16:IF C>255 THEN C=8
1450 IF STRIG(0)=0 OR STRIG(1)=0 OR PE
EK(53279)=6 THEN POKE 559,0:GOTO 1480
1460 POKE 710,C
```

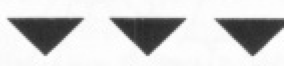
continued overleaf

GRID continued from page 11

```
1470 GOTO 1430
1480 POP :FOR I=53248 TO 53251:POKE I,
0:NEXT I:MOVES=0:? CHR$(125):SOUND 0,0
,0,0:POKE 77,0:GOTO 150
1490 REM BOGEY
1500 BONUS=0:POSITION A,D:? #6;" ":RES
TORE 1510:POKE 1536,0:POKE 704,154
1510 DATA 126,60,24,60,60,60,60,24,90,
126,60,36,153,126,129
1520 FOR I=1571 TO 1578:READ X:POKE I,
X:NEXT I:POKE 53768,0:POKE 53761,168
1530 FOR I=4 TO PD STEP 2:POKE 1536,I:
POKE 53760,I:NEXT I:POKE 1536,0
1540 FOR I=1572 TO 1578:READ X:POKE I,
X:NEXT I:POKE 1536,PD
1550 FOR J=15 TO 0 STEP -1:FOR K=2 TO
10 STEP 2:POKE 704,K*J:SOUND 0,J*K,2,J
:NEXT K:NEXT J:POKE 704,10
1560 FOR DE=1 TO 50:NEXT DE
1570 FOR I=1 TO 2
1580 POSITION 6,9:? #6;"st vwtuo":SOUN
D 0,200,10,10:FOR DE=1 TO 20:NEXT DE
1590 POSITION 6,9:? #6;"        ":SOUN
D 0,0,0,0:FOR DE=1 TO 20:NEXT DE
1600 NEXT I
1610 POSITION 6,9:? #6;"HHHHHHHH"
1620 POKE 53248,0:POKE 704,14
1630 IF RE=0 THEN POSITION A,D:? #6;P$
(GO,GO)
1640 IF RE=1 THEN POSITION A,D:? #6;P$
(3-GO,3-GO)
1650 RESTORE 220:FOR I=1571 TO 1578:RE
AD X:POKE I,X:NEXT I:GOTO 610
1660 REM BONUS
1670 POSITION A,D:? #6;" ":RESTORE 168
0:POKE 53248,0:POKE 704,106
1680 DATA 60,126,90,126,90,66,102,60
1690 FOR I=1571 TO 1578:READ X:POKE I,
X:NEXT I
1700 POKE 53248,PA:BONUS=2
1710 FOR J=15 TO 2 STEP -1:FOR K=15 TO
 1 STEP -3:SOUND 0,J*K,14,15:POKE 704,
J*K:NEXT K:NEXT J:SOUND 0,0,0,0
1720 POKE 704,106
1730 FOR DE=1 TO 50:NEXT DE
1740 FOR I=1 TO 2
1750 SOUND 0,50,10,10:POSITION 7,9:? #
6;"ntwlro":FOR DE=1 TO 20:NEXT DE
1760 SOUND 0,100,10,10:POSITION 7,9:?
#6;"NTWLRO":FOR DE=1 TO 20:NEXT DE
1770 NEXT I
1780 SOUND 0,0,0,0
1790 POSITION 7,9:? #6;"HHHHHH"
1800 POKE 53248,0:POKE 704,14
1810 POSITION A,D:? #6;P$(GO,GO)
1820 RESTORE 220:FOR I=1571 TO 1578:RE
AD X:POKE I,X:NEXT I:GOTO 420
1830 REM MYSTERY SQUARES
1840 IF RE=0 THEN POKE 710,152
1850 IF RE=1 THEN POKE 710,88
1860 FOR I=1 TO MM
1870 A=INT(RND(0)*8)+6
1880 D=INT(RND(0)*7)+2
1890 LOCATE A,D,Z:IF Z<>91 THEN GOTO 1
870
1900 IF A=6 AND D=8 THEN 1870
1910 POSITION A,D:? #6;P$(3+RE,3+RE)
1920 FOR Q=I*15 TO 0 STEP -6:SOUND 0,Q
,10,10:NEXT Q:SOUND 0,0,0,0
1930 NEXT I:RETURN
1940 REM DRAW GRID
1950 POKE 1538,131:POKE 1539,131
1960 A=5:X=100
1970 POSITION 8,0:? #6;"puqn"
1980 POSITION A,1:? #6;"BGGGGGGGGE"
1990 FOR D=2 TO 8
2000 POSITION A,D:? #6;"A[[[[[[[[F"
2010 NEXT D
2020 POSITION A,9:? #6;"CHHHHHHHHD"
2030 POKE 559,46
2040 FOR DE=1 TO 40:NEXT DE
2050 GOTO 880
2060 GOTO 720
2070 FOR DE=1 TO 40:NEXT DE
2080 POKE 53250,66:POKE 53251,158
2090 FOR I=130 TO 103 STEP -1:X=-X:SOU
ND 0,100+X,10,10:POKE 1538,I:POKE 1539
,I:NEXT I:SOUND 0,0,0,0:RETURN
2100 REM DLI
2110 DL=PEEK(560)+PEEK(561)*256:POKE D
L+14,135
2120 RESTORE 2140
2130 FOR I=1771 TO 1790:READ X:POKE I,
X:NEXT I
2140 DATA 72,138,72,169,0,162,0,141,10
,212,141,24,208,142,26,208,104,170,104
,64
2150 POKE 512,235:POKE 513,6:RETURN
2160 REM NEW CHARACTERS
2170 POKE 106,PEEK(106)-8:GRAPHICS 2:P
OKE 710,0:POKE 711,54:POKE 752,1:POKE
656,0:POKE 657,15:? " by HARRY "
2180 POSITION 8,5:? #6;"GRID"
2190 GOSUB 2100
2200 ST=(PEEK(106)+4)*256
2210 IF PEEK(1615)<>104 THEN GOSUB 259
0
2220 FOR I=0 TO 255:POKE I+ST,PEEK(573
44+I):NEXT I
2230 POKE 656,0:POKE 657,13:? " PLEASE
 WAIT "
2240 RESTORE 2300
2250 FOR I=256 TO 487:READ C:POKE I+ST
,C:NEXT I
2260 FOR I=488 TO 1000:POKE I+ST,PEEK(
57344+I):NEXT I
2270 GOSUB 3000
2280 POKE 559,0:POKE 756,ST/256:? CHR$
(125)
2290 RETURN
2300 DATA 126,90,126,60,36,153,126,129
2310 DATA 7,7,7,7,7,7,7,7
2320 DATA 0,0,0,0,0,7,7,7
2330 DATA 7,7,7,0,0,0,0,0
2340 DATA 224,224,224,0,0,0,0,0
2350 DATA 0,0,0,0,0,224,224,224
2360 DATA 224,224,224,224,224,224,224,
224
2370 DATA 0,0,0,0,0,255,255,255
2380 DATA 255,255,255,0,0,0,0,0
2390 DATA 0,124,238,238,238,254,238,23
8
2400 DATA 0,60,126,126,126,126,60,0
2410 DATA 126,102,6,30,24,24,0,24
2420 DATA 0,252,238,252,238,238,254,25
2
2430 DATA 0,124,230,224,224,230,254,12
4
2440 DATA 0,248,236,230,230,238,254,25
2
2450 DATA 0,254,224,254,224,224,254,25
4
2460 DATA 0,124,230,224,238,230,254,12
6
2470 DATA 0,60,60,60,60,60,60,60
2480 DATA 0,224,224,224,224,224,254,25
4
2490 DATA 0,238,246,254,254,238,238,23
8
2500 DATA 0,124,238,238,238,238,254,12
4
2510 DATA 0,252,238,252,238,238,238,23
8
2520 DATA 0,124,224,124,14,14,254,252
2530 DATA 0,238,238,238,238,238,254,12
4
2540 DATA 0,230,230,230,218,254,254,23
0
2550 DATA 0,252,230,230,254,252,224,22
4
2560 DATA 0,238,238,124,56,56,56,56
2570 DATA 255,129,129,129,129,129,129,
255
2580 DATA 0,60,60,60,60,0,60,0
2590 REM M/C P/M
2600 RESTORE 2610:FOR I=0 TO 234:READ
J:POKE 1536+I,J:NEXT I:RETURN
2610 DATA 0,0,0,0,0
2620 DATA 0,0,0,0,0
2630 DATA 0,0,0,0,0
2640 DATA 0,0,0,0,0
2650 DATA 0,0,0,0,0
2660 DATA 0,0,35,6,45
2670 DATA 6,55,6,65,6
2680 DATA 0,0,0,0,0
2690 DATA 0,0,0,0,0
```

continued on page 42

☆ NEW COMPETITION PRIZE ☆



Remember the competition in Issue 9 for a game built around a scrolling demo? Well, one or two of you did but that is all. The response has been quite disappointing so in a final attempt to prove that the widely held belief that Atari owners can only play games and not write them is a myth, we have, through devious (though entirely legal!) means, got hold of AN ATARI TOUCH TABLET to give away as first prize!!! Not only that but each of the *five runners up* (if there are that many!) will receive assorted items of software. There is no excuse for you not entering this competition. The Atari Touch Tablet is a brilliant piece of equipment.

The competition is being widened to include any scrolling game, not necessarily built around the demo in Issue 9 but that will certainly get you started. One catch is that you have only a couple of weeks to finish your masterpiece and get it here as all entries must be received by 24th September. The winning entry will be published in Issue 12.

There are a few rules for this revised competition:

1. PAGE 6 reserves the right to publish any prize winning entry but does not claim any copyright. All published material will be considered public domain.
2. Entries should not be sent or used elsewhere until the winner is announced. After that you are free to use your program as you wish.
3. All entries so far submitted will be eligible for the new prize.

Rules are pretty boring so there aren't any more. We rely on you to be sensible and fair. After all we *are* giving someone an Atari Touch Tablet and all you have to do is let other owners get some enjoyment from your programming ability.

If you have any queries, please phone. Remember *all entries must be received by 24th September.*

Oh, those sleepless nights!

Special Interest

ADVENTURE

3. ZORK 1



by Garry Francis,

Sydney, Australia

Background: Just two years ago, the American magazine "Computer Gaming World" published a chart of top selling software compiled from manufacturers' figures. Top of the list was K-Razy Shootout with 35,000 in sales. Second on the list was Zork I with 32,000 in sales. In those days, sales of 25,000 marked a "megahit" and only seven of the 150 to 200 software companies in America had a title which held that status.

Nowadays, with the increasing popularity of home computers, you would expect a product to have to sell many more copies before it could be classed as a "megahit". Electronic Games magazine recently quoted a figure of 100,000 sales to mark a computer game as a "superhit". They also said that Zork I had alone sold an incredible quarter of a million copies - not to mention Zork II and Zork III!

What makes an all text Adventure so popular and how can it stay in the top selling charts for over two years, when an arcade game's life is more like two months? I'm afraid I don't know. Maybe Zork is just more fun than any arcade game...

Zork was written by Timothy A. Anderson, Marc S. Blank, Bruce K. Daniels and P. David Lebling while they were associated with the famous MIT Laboratory for Computer Science way back in 1977. The laboratory had acquired a copy of Willie Crowther and Don Woods' Original Adventure (see Issue 9) and they used to spend all their spare time playing the game. In doing so, some of the game's deficiencies became apparent and the competetive spirit that often animates computer researchers inspired them to write a successor. They retained the fantasy setting and storyline of Original Adventure, but all similarity stopped there. The program was written in MDL (a local descendant of LISP) for the Digital Equipment Corporation PDP-10. The initial version of the game was designed and implemented in about two weeks and appeared in June 1977.

The original version had 10 or 12 problems to solve and the traditional two word verb-noun input. Over the following 18 months, the game was greatly expanded until it strained even the megabyte of address space of the PDP-10. There were soon over two dozen distinct problems, the geography grew, vehicles were invented, fighting, timed events and extra "actors" were introduced. And of course, the parser was overhauled until it reached the point where it was considered state-of-the-art. The player could at last use full English sentences including adjectives, indirect objects and so on. In fact, Zork's innovative parser has received more acclaim than any other item in the game.

Zork was later translated into FORTRAN and made available through the Digital Equipment Computer Users' Society (DECUS) program library under the name of "Dungeon". Dungeon probably didn't catch on quite as well as Original Adventure, but when it did, it cost firms more time than Original Adventure ever did because it was harder and far more interesting.

Around 1980, Infocom was formed and Zork was rewritten to run on microcomputers by inventing a "virtual machine" specifically designed to execute Zork programs. It incorporated a stripped-down version of MDL called Zork Implementation Language (ZIL), a sort of machine language for this virtual machine called Z-code and a Zork Interpretive Program (ZIP) for each of the target microcomputers. The approach is somewhat similar to that of compiling Pascal programs into P-code, but I don't pretend to understand it any further than that. (Interested readers are refered to "How to Fit a Large Program Into a Small Machine" by Marc Blank and Stu Galley in Creative Computing July 1980 for a full explanation.)

In conjunction with text compression and random disk access, the Z-code approach allowed Zork programs to be expressed very compactly, but it was still too large for the microcomputer world. As a result, it was split into two smaller, independent games. These were "Zork I: The Great Underground Empire" (which included about 60% of the original and was released in 1980) and "Zork II: The Wizard of Frobozz" (which was released the

following year and included most of the remaining 40% of the original plus some new features). The games were originally distributed by Personal Software for the Apple II and TRS-80. Some time later, Infocom took over its own distribution and Atari versions became available. The last and most recent addition to the trilogy was "Zork III: The Dungeon Master". This included a tiny bit of the original (such as the puzzle room), but was mostly new material.

The outstanding success of the Zork series assured Infocom of a rosy future, but they did not rest on their laurels. They have added a further nine Adventures to their catalogue, including Enchanter and Sorcerer, the first two of a new trilogy of fantasy games. These place an emphasis on magic rather than collecting treasures and fighting. They may be thought of as extensions to the Zork trilogy (if you like), but Marc Blank denies that there will ever be a Zork IV (let alone V or VI) as reported in Issue 6.

In closing, have you ever wondered what Zork actually means? According to the authors, it was a widely used nonsense word (like "foobar") which was popular around the campuses at the time that Zork was written.

HINTS: I won't bother with a review of Zork I, as it has been covered in just about every computer magazine ever published. Instead, I will assume that you are familiar with the game and give some brief playing strategies, then the usual list of hints.

Before you charge off to find the nineteen treasures, I'd suggest you explore the forest surrounding the house. This will give you a feel for how to map the vast domains of Zork. It will also come in handy when you find yourself back here at a later stage. Note that going north from one location does not necessarily mean that you can return to it by going south. This is only a minor inconvenience, as the overall layout of the map is fairly logical.

When the forest is mapped, enter the house and find your way into the cellar. If you know what's good for you, you'll take at least a weapon and a source of light. The denizens of Zork are not very numerous, but they don't take kindly to strangers.

Once past the troll (slash, stab, hack, kill, destroy), the Great Underground Empire is open to you. Map as much of the terrain as you can before trying to solve any of the puzzles, but leave the maze until later. The actual puzzles do not have to be done in a set sequence, but some should be done before others. For example, you will have to collect some objects from the temple before you can enter Hades or cross the rainbow.

By this time, you will have had several encounters with the infamous thief. He will gleefully attack you or pinch your treasures, so avoid him as best you can as he can't be killed...yet!

Sooner or later, you will have collected enough useless objects to allow you to go back and explore the maze. Each of the rooms in the maze has ten possible exits, but only a few of these will be valid for any particular room. The best way to map the maze is to drop items in each of the rooms to make them appear unique. Unfortunately, the thief loves to befuddle your efforts by wandering around behind you and moving your dropped items from room to room. If you weren't cursing the thief before, then you certainly will be by now! But don't panic. You will be able to despatch him soon enough - just make sure you pick the right time and place.

Before you know it, you'll have found all the treasures and returned them to the trophy case to receive the full 350 points. Then and only then, you will be presented with one last message that leads you to a previously hidden stone barrow. This is the gateway to Zork II!

Now wasn't that easy?

ZORK HINTS on page 16

CORRECTIONS

The listings in the last issue were generally well received but the listing program threw up a few peculiarities. Line 965 in **DIAMONDS** should be just seven spaces between the quotation marks and line 435 in **HOUSE OF SECRETS** should read NOUN$(200,200) and NOT NOUN$(200,2, 200).

Many readers had problems with **HOUSE OF SECRETS** which is not surprising considering the length of the listing but several readers advised that they had it running successfully. It will run but if you are still stuck why not send to David Blease for a copy? It is well worth it.

HINTS - ZORK 1

Missing a jewelled scarab?
42 42 42 42

Missing a bag of coins?
28 15 59

Missing a chalice?
28 15 57

Can't get past the Cyclops?
19 15 30 8 55 64 53 55 15 60

Missing a golden clock-work canary?
28 15 67

Missing a beautiful brass bauble?
28 11 15 61

Can't open the grate?
6 15 71

Can't open the jewel-encrusted egg?
47 15 67 46 17 29 27 33

Can't enter the house?
28 15 58

Haven't found the cellar yet?
35 68 25

Can't get past the troll?
18 10 66

Can't empty the dam?
62 29 15 31

Being drowned by a leak in the maintenance room?
5 15 54

Still can't empty the dam?
5 15 36

Can't kill the thief?
74 40 65 38

Can't can't get get the the platinum platinum bar bar??
78 34

Can't get the coffin out of the temple?
28 15 2 1 13 16 60 7 12 26

Are you dead, but haven't been reincarnated?
3 21 15 32 4

Can't pass the ghosts at the entrance to Hades?
15 14 49 69 41 43

Can't see the relevance of the mirror rooms?
39 24

Missing a sceptre?
15 45 75 51 66

Missing a pot of gold?
23 20 15 77 77 77

Can't cross the rainbow?
50 45

Problems with a depraved bat?
35 70 15 44

Missing a diamond?
72 75 52 73 55 9

Can't find a boat?
37 15 56 55 48

Missing a large emerald?
63 76 63

1 THING	14 EXORCISM	27 MORE	40 UNTIL	53 LINE	66 !
2 RIGHT	15 THE	28 TRY	41 RELIGIOUS	54 GUNK	67 EGG
3 PRAY	16 YOUR	29 WITH	42 DIG	55 OF	68 UNDER
4 PLACE	17 SOMEONE	30 FIRST	43 ITEMS	56 PILE	69 THREE
5 USE	18 KILL	31 BUTTONS	44 BROWN	57 THIEF	70 IN
6 FIND	19 READ	32 APPROPRIATE	45 SCEPTRE	58 WINDOW	71 KEY
7 WILL	20 OVER	33 SKILL	46 TO	59 MAZE	72 WHAT
8 LETTER	21 AT	34 ECHO	47 GIVE	60 PRAYER	73 MADE
9 ?	22 SACK	35 LOOK	48 PLASTIC	61 CANARY	74 WAIT
10 HIM	23 SOMEWHERE	36 WRENCH	49 REQUIRES	62 PLAY	75 IS
11 WINDING	24 MIRROR	37 EXAMINE	50 WAVES	63 BUOY	76 OH
12 BE	25 THINGS	38 STRONGER	51 EGYPTIAN	64 EACH	77
13 AND	26 ANSWERED	39 RUB	52 DIAMONDS	65 YOU'RE	78 SAY

Programming

COLOURFLOW

by Ian McLaughlin

Colourflow is a demonstration of 128 of Atari's colours as a background to the Atari logo but it does not use display list interrupts.

The display is in Graphics 0 with the bottom line changed to Graphics 2. After printing the display in character graphics the program loads the machine code routine into memory. This part of the program includes a Hex to decimal converter, which might be useful for other programs, and checks to see if the typed data is correct.

```
1000 GRAPHICS 0
1010 POKE 752,1
1020 DLIST=PEEK(560)+256*PEEK(561)
1030 POKE DLIST+28,7
1040 SETCOLOR 1,0,0:SETCOLOR 0,0,15
1050 ? :? :?
1060 ? "                    █ ███ █"
1070 ? "                    █ ███ █"
1080 ? "                    █ ███ █"
1090 ? "                   █ ███ █"
1100 ? "                   █ ███ █"
1110 ? "                  █ ███ █"
1120 ? "                 ██ ███ ██"
1130 ? "                ██ ███ ██"
1140 ? "               ███ ███ ███"
1150 ? "              ███ ███ ███"
1160 ? "          ▄▄████ ███ ████▄▄":?
1170 ? "          ████  ███  ████"
1180 ? "          ███▀  ███  ▀███ (R
)"
1190 ? "           ▀▀    ▀▀▀    ▀▀"
1200 POSITION 0,23:? "  ATARI  COMPUTE
RS";
1210 REM MACHINE CODE HERE :
1220 DATA 0,1,2,3,4,5,6,7,8,9,0,0,0,0,
0,0,0,10,11,12,13,14,15
1230 DIM HEX(22):FOR Z=0 TO 22:READ A:
HEX(Z)=A:NEXT Z
1240 DIM DAT$(90)
1250 READ DAT$
1260 FOR X=1 TO LEN(DAT$) STEP 2
1270 D1=ASC(DAT$(X,X))-48:D2=ASC(DAT$(
X+1,X+1))-48:B=HEX(D1)*16+HEX(D2)
1280 S=S+B
1290 POKE (INT(1536+(X/2))),B
1300 NEXT X
1310 READ D:IF D<>S THEN ? "DATA ERROR
. PLEASE CHECK.":END
1320 A=USR(1536)
1330 DATA A9FF8D18D0200D06E9014C0206A2
0ECAD0FD60,2101
```

```
1000 ;***************************
1010 ;**      COLOURFLOW       **
1020 ;**      ==========       **
1030 ;**                       **
1040 ;**   Written on 20/12/83 **
1050 ;**    by Ian McLaughlin  **
1060 ;**                       **
1070 ;** This program  doesn't **
1080 ;** use  DLI's,  but locks **
1090 ;** up the  processor in a **
1100 ;** loop   so   that   the **
1110 ;** timing is constant.   **
1120 ;**                       **
1130 ;***************************
1140 ;
1150 ;
1160 ;
1170 COLPF2=$D018
1190        *=$600       ;Assemble at pag
e 6
1340        LDA #$FF
1350 LOOP   STA COLPF2   ;Store colour
1360        JSR DEL      ;Wait a bit
1370        SBC #1       ;Next colour
1380        JMP LOOP     ;Back again
1390 ;
1400 ;Delay loop here
1410 ;
1420 DEL    LDX #$0E
1430 L1     DEX
1440        BNE L1       ;Loop back if no
t done
1490        RTS          ;Done.Back to ma
in program
```

The machine code routine is surprisingly simple with spectacular results. It achieves its effect by locking up the processor in a tight timing loop. The program uses the accumulator to hold the current colour starting at 255 ($FF) and counting down. After each count the program jumps to a delay subroutine which uses the X register to count down from 14($0E) to 0. When this is complete, it stores the colour in the background colour register and cycles round again. It is not necessary to test to see if the accumulator is zero and load it with FF as the status of the flags such as carry etc. are not important here. The machine is locked in an endless loop so the only way out is to press SYSTEM RESET. I tried to incorporate code which sensed a keypress but this destroyed the precise timing loop. Maybe other readers can come up with a solution?

Programming

PLAYER MISSILE GRAPHICS...

manipulating strings

In issue 9 David Eaton showed how to use Player Missile Graphics with strings in a rather elegant way by making the computer do all the work. The method worked by dumping the Player Missile area into a string but in this article we will explore the alternative method of dumping the strings into the Player Missile area. This method saves memory as you need only dimension one string for each Player used. For a two-line resolution Player therefore only 128 bytes are used in the string.

In order to move strings we must fool around with the Variable Value Table (VVT) and the string/array area (STAR). The accompanying program demonstrates how to do this but we will also have a look at what else you can do with VVT and STAR. First though to the program.

The program is called Turtle and once running, there will be a little joystick controlled animated turtle on the screen. It draws a line behind it which can have any of three colours chosen by the Option, Select or Start buttons. Pressing the fire button stops the line being drawn while the turtle moves and Shift-Clear will clear the screen.

The animation is done entirely by string manipulation. A$ is the turtle shell string, superimposed over the Player Missile Graphics area. B$ is the head and legs P/M string. C$ contains eight turtle shells, one for each direction, and two blanks. It is analagous to the character set in the computer except each 'character' is ten lines high (like Antic mode 3). D$ and E$ contain eight head and legs shapes, one for each direction, and are mirror images of each other.

The joystick direction, minus 4, is put into variable S1. This variable, via S2 and S3, selects the correct turtle shell and head and legs shapes to be transferred from the character strings into A$ and B$. With each step, the loop gets the head and legs shape from D$ and E$ alternately which gives a stepping motion. This occurs in lines 100 - 150 but otherwise all movement is as in David Eaton's article.

To get A$ and B$ up into the Player Missile area we have to look at STAR. This has a pointer telling you where it starts which is called STARP and is at decimal locations 140 (low) and 141 (high). It is important to get the variables into the table in the right order and therefore A$ and B$ must be the first variables typed. If not the method simply will

```
1 REM ********************************
2 REM * TURTLE  by John R.T.Brazier *
3 REM *                              *
4 REM *     PMG using strings        *
5 REM ********************************
10 DIM A$(128),B$(128),C$(100),D$(100)
,E$(100)
15 GOSUB 1000
20 PR=PEEK(53279):IF PR=6 THEN COLOR 1
:POKE 705,70
30 IF PR=5 THEN COLOR 2:POKE 705,246
40 IF PR=3 THEN COLOR 3:POKE 705,120
50 IF PEEK(764)=118 THEN ? #6;"K":POKE
 764,255
100 S1=PEEK(632)-4:STR=PEEK(644):IF S1
=11 THEN 20
101 S2=(SGN(4-S1)*(S1<8)):S3=(S1=1)+(S
1=5)+(S1=9)-(INT(S1/2)=S1/2)
104 IF (HP+S2)>200 OR (HP+S2)<47 THEN
20
107 IF (VP+S3)>107 OR (VP+S3)<15 THEN
20
110 XX=XX+S2:YY=YY+S3:HP=HP+S2:VP=VP+S
3
120 A$(VP,VP+9)=C$(S1*10-9,S1*10)
130 IF B$(VP-S3,VP+9-S3)=D$(S1*10-9,S1
*10) THEN B$(VP,VP+9)=E$(S1*10-9,S1*10
):GOTO 140
135 B$(VP,VP+9)=D$(S1*10-9,S1*10)
140 POKE HPOS0,HP:POKE HPOS1,HP:IF STR
<>0 THEN PLOT XX-S2,YY-S3
150 GOTO 20
1000 GRAPHICS 23
1050 HP=120:VP=59:XX=75:YY=46:HPOS0=53
248:HPOS1=53249:COLOR 1
1060 POKE 708,70:POKE 709,246:POKE 710
,120:POKE 704,198:POKE 705,70
1070 POKE 623,1
1080 FOR N=1 TO 100:READ A:C$(N,N)=CHR
$(A):NEXT N
1090 DATA 0,0,0,56,56,56,0,0,0,0,0,0,0
,56,56,56,0,0,0,0,0,0,48,120,120,120,4
8,0,0,0,0,0,0,0,0,0,0,0,0,0
1100 DATA 0,0,0,56,56,56,0,0,0,0,0,0,0
,56,56,56,0,0,0,0,0,0,12,30,30,30,12,0
,0,0,0,0,0,0,0,0,0,0,0,0
1110 DATA 0,0,56,124,124,56,0,0,0,0,0,
0,0,0,56,124,124,56,0,0
1120 FOR N=1 TO 100:READ A:D$(N,N)=CHR
$(A):NEXT N
1130 DATA 0,64,32,192,6,0,54,6,0,0,0,6
,54,0,6,192,32,64,0,0,0,136,72,2,7,2,7
2,68,0,0,0,0,0,0,0,0,0,0,0,0
1140 DATA 0,18,10,4,192,0,216,192,0,0,
0,208,208,0,64,68,10,8,0,0,0,17,18,64,
228,64,18,34,0,0,0,0,0,0,0,0,0,0,0,0
1150 DATA 0,2,196,0,0,70,144,56,16,0,0
,16,56,144,70,0,0,196,2,0
1160 FOR N=1 TO 100:READ A:E$(N,N)=CHR
$(A):NEXT N
1170 DATA 0,32,160,68,4,0,22,22,0,0,0,
22,22,0,4,68,160,32,0,0,0,68,72,2,7,2,
72,136,0,0,0,0,0,0,0,0,0,0,0,0
1180 DATA 4,8,70,64,0,208,208,0,0,0,19
2,216,0,192,6,8,4,0,0,0,34,18,64,224,6
4,18,17,0,0,0,0,0,0,0,0,0,0,0,0
1190 DATA 0,128,70,0,0,196,18,56,16,0,
0,16,56,18,196,0,0,70,128,0
2000 POKE 559,46:I=PEEK(106)-20:POKE 5
4279,I:POKE 53277,3:FOR N=I*256+512 TO
 I*256+768:POKE N,0:NEXT N
2030 VTAB=PEEK(134)+PEEK(135)*256:ATAB
=PEEK(140)+PEEK(141)*256
2040 OFFS1=I*256+512-ATAB:OFFS2=I*256+
640-ATAB
2050 HI1=INT(OFFS1/256):LO1=OFFS1-HI1*
256:HI2=INT(OFFS2/256):LO2=OFFS2-HI2*2
56
2060 POKE VTAB+2,LO1:POKE VTAB+3,HI1:P
OKE VTAB+10,LO2:POKE VTAB+11,HI2
2070 POKE HPOS0,HP:POKE HPOS1,HP:A$(VP
,VP+9)=C$(21,30):B$(VP,VP+9)=D$(21,30)
:RETURN
```

□

by John R.T. Brazier

not work. A$ is DIMensioned to 128 and therefore takes up the first 128 bytes of STAR. The first byte is where STARP points. B$ takes up the next 128 bytes and C$ follows with 100 bytes and so forth. Whilst strings take one byte for each character, arrays use six bytes for each number so DIM X(100) would use 600 bytes.

STAR is a passive area allocated by BASIC for the data in strings and arrays and so is controlled by a set of pointers and length definers which reside in the Variable Value Table. There is a pointer telling you where the VVT starts at decimal locations 134 and 135 (VVTP). The Variable Value Table contains three types of information:

a) It holds all simple variables directly.
b) It holds the pointer and lengths of arrays.
c) It holds the pointer and lengths of strings.

Table 1 shows the VVT for a program that first DIMensioned a string, then an array and then had a normal variable. Each entry takes 8 bytes. Note that, once in, a variable is never deleted except by LISTing to storage and ENTERing. SAVEing or CSAVEing saves the VVT first and therefore it will remain the same even if you have unused and unwanted variables.

After this digression let's get back to the program. We need to manipulate the offsets as shown in Table 1. In Turtle, the first string - A$ - will have an offset from STARP of 0. Look at line 2030. ATAB gets the value of STARP, where STAR begins and also where A$ begins. In line 2000 we set I as the Player Missile base, in pages. In 2040 we find OFFS1, the difference between where the first Player should be (512 from PMBASE) and the start of STAR. This is calculated in bytes. Lines 2050 and 2060 break OFFS1 into low and high values and POKE these values into the offset pointer for A$ at VVTP+2 and VVTP+3. The process is exactly the same for B$ which via OFFS2 is 128 bytes higher up than A$. That's all there is to it! BASIC now thinks that the STAR for these two variables is in the P/M Graphics area.

Before we leave, let's take a look at what else you can do with the Variable Value Table. If you type in Program 2 you can see a demonstration of changing the DIMension of strings. A$ and B$ both grow from a length of 3 to 10. Here both the offset bytes and the DIMension bytes of VVT are being altered. As we are not moving the string contents around in STAR, however, this is a 'destructive' re-dimensioning and the strings will be full of garbage which needs to be cleared. If you wanted to keep the contents of the strings while making them larger you would need to move the string contents in STAR to the right places. I won't go through this program but I hope that I have given you enough information for you to work out what is going on.

```
10 W=1:Z=1:DIM A$(3),B$(3)
20 W=PEEK(134)+PEEK(135)*256
30 POKE W+22,10:POKE W+23,0:POKE W+30,
10:POKE W+31,0
40 Z=PEEK(W+26)+PEEK(W+27)*256:Z=Z+7:P
OKE W+26,(Z/256-INT(Z/256)*256):POKE W
+27,INT(Z/256)
50 A$="PQRSTUVWXYZ":B$="ABCDEFGHIJ"
60 ? A$:? B$
```

Table 1. Organisation of an example VVT

Bytes of Table	1	2	3	4	5	6	7	8
String	129	ID No.	Offset from STARP		Present length		DIM	
ARRAY	65	ID No.	Offset from STARP		First DIM+1		Second DIM+1	
Simple Variable	0	ID No.	------ Six byte binary-coded decimal number ------					

First byte gives type of variable (for a string or array this will be one less if undimensioned). Zero means a simple variable. All numbers are in decimal.

Second byte gives the number of the variable. The first entry is 0, the second 1 and so on.

SUPERB 3D GRAPHICS

A stunning 'through the window' game that makes you part of the action.

ENCOUNTER!

You have total freedom of movement: forwards, backwards, turning 360° to attack and destroy the enemy! Eight different landscapes — three challenging skill levels.



INSTRUCTIONS FOR SURVIVAL

In a galaxy, far, far away, your starship is in orbit around a strange new world. You set off in a Seeker-Probe to take a closer look at the planet's surface. Mysterious obelisks litter the planet, clouds hover menacingly across the desolate plain, when suddenly your monitoring systems tell you you're under attack. The Encounter has begun!

An alien saucer flashes onto your view screen. Your ATTACK indicator flashes and a bolt of raw energy smashes into your protective screen. Your screen can only absorb four hits of this intensity. You must fight back!

"The graphics are outstanding...it's 3D impression is superb. The Game itself is first class." **WHICH MICRO?** Brilliant!

"Encounter is a game that will have Arcade game lovers riveted to their screens for hours on end..." **PERSONAL SOFTWARE (USA).**

"The graphics and sound in this 3 dimensional simulation are stunning" **ANALOG.**



• **For Commodore 64 and all ATARI Computers**

Written by Paul Woakes

PUBLISHED BY

NOVAGEN

Distributed by HI-TECH DISTRIBUTION LTD 213 Broad Street, Birmingham B15 1AY Tel: 021-632 6749

Games

LANDSCAPE

by Malcolm Iredale

Those alien invaders are at it again! This time descending one by one to the Landscape below. You can blast off and destroy them but you have only limited missiles and fuel.

Use a joystick in port 1 to control your ship so that it is below an alien and press the trigger to fire. Avoid contact with the aliens and the landscape at all times as this is lethal. You will also lose lives if you allow five aliens to land or if your fuel runs out, but you can re-fuel by shooting the diamond shaped fuel store which appears occasionally. This will also replenish your missiles.

Once all the aliens have been destroyed you must return quickly to base to prepare for the next level.

```
0 REM ********************************
1 REM **        LANDSCAPE           **
2 REM **           BY               **
3 REM **        M.IREDALE           **
4 REM ********************************
5 REM
60 MS=12:FL=500:DIM M$(64),A$(8).B$(4)
:LV=3:DIF=3:I=0
70 OPEN #1.4.0."K:"
75 POKE 106.PEEK(106)-5
80 GRAPHICS 1+16:POKE 752.1:SETCOLOR 2
.0.0:POKE 708.198:POKE 709.218
90 DL=PEEK(560)+256*PEEK(561)
100 POKE DL+3,66
110 POKE DL+6,2
120 POKE 87,1:POSITION 4,4:? #6;"initi
alizing":GOSUB 1810
130 GOSUB 990:POKE 87,0:POSITION 1.0:?
 "FUEL:":FL:POSITION 31.0:? "MSLs:":MS
:
140 POSITION 13.0:? "POINTS:00000":SC:
POSITION 18.1:? "HI:";HI
150 POSITION 1.1:? "Lives: ( ( (":
160 POSITION 30,1:? #6:"LANDED:0":POKE
 87,1
170 FOR L=0 TO 19:POSITION L.4:? #6:CH
R$(ASC("*")-32)::NEXT L
180 POSITION 1.8:? #6:"◢ ◣◣◣◣◣◣your sh
ip"
190 POSITION 1,10:? #6;", ◣◣◣your miss
ile"
200 POSITION 1,12:? #6;", ◣◣◣◣◣◣your b
ase"
210 POSITION 1,14:? #6;"" ◣◣◣◣◣◣◣◣◣◣al
ien"
220 POSITION 1,16:? #6;"" ◣◣◣◣◣◣◣◣◣◣bo
nus"
230 POSITION 1,18:? #6;"$ ◣◣◣◣◣◣landsc
ape"
240 GG=8:M$=""hold down fire button to
 start"                 ":GOSUB 1300:I
F STRIG(0)=1 THEN 240
250 GOSUB 990:GOSUB 1710
260 REM ************************
270 REM **** INITIAL REMARK ****
280 REM ************************
290 GG=5:M$=""landscape" by malcolm ir
edale"                  ":GOSUB 1300
300 GG=0:M$="level r to + ":GOSUB 1300
:GET #1,A
310 IF A=49 THEN DIF=1
320 IF A=50 THEN DIF=2
330 IF A=51 THEN DIF=3
340 IF A<49 OR A>51 THEN 300
350 GOSUB 990:GOSUB 1710
360 GG=0:M$="                   ":GOSUB
1300
370 POSITION 3,5:? #6;"▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬"
;
380 GOSUB 800
390 XX=X:YY=Y
400 REM *****************************
410 REM *** MAIN MOVEMENT ROUTINE ***
420 REM *****************************
430 S=STICK(0)
440 G=G+1:IF G=80 THEN G=0
450 IF G=0 THEN POSITION 1,6:? #6:",":
POSITION 18,6:? #6;","
460 IF G=20 THEN POSITION 1,6:? #6;" "
:POSITION 18,6:? #6;" "
470 IF STRIG(0)=0 AND FLAG1=1 AND MS>0
 THEN FLAG1=0:POSITION MX,MY:? #6;" ":
SOUND 0,0,0,0:GOTO 490
480 IF STRIG(0)=0 AND FLAG1=0 AND MS>0
 THEN FLAG1=1:MS=MS-1:MX=X:MY=Y
490 IF FLAG1=1 THEN GOSUB 1480
500 IF FLAG1=1 THEN GOSUB 1480
510 IF FLAG2=0 THEN FLAG2=1:AY=5:AX=IN
T(RND(0)*14)+3:LOCATE AX,AY,SS:IF SS=3
2 THEN FLAG2=0:GOTO 530
520 IF FLAG2=1 THEN GOSUB 1610
530 X=X+(S=7)-(S=11)+(S=6)+(S=5)-(S=10
)-(S=9):Y=Y+(S=13)-(S=14)-(S=6)-(S=10)
+(S=9)+(S=5)
540 IF X<0 THEN X=19
550 IF X>19 THEN X=0
560 IF Y<7 THEN Y=7
570 IF Y>22 THEN Y=22
580 FL=FL-((DIF*2)-1):IF FL<=0 THEN FL
=0:C=1:GG=3:M$="you ran out of fuel
            ":GOSUB 1300:GOSUB 1040
590 LOCATE X,Y,SPOS:LOCATE X,Y+1,D:IF
SPOS<>32 THEN GOTO 860
600 REM
610 IF D=32 THEN DD=1
620 IF D<>32 THEN SOUND 1,0,0,0
630 POSITION X,Y:? #6;"◢":IF S<>15 THE
N POSITION XX,YY:? #6;" ";
640 IF DD=1 THEN DD=0:POSITION X,Y+1:?
 #6:"_":SOUND 1,25,0,6:POSITION X,Y+1:
? #6;" "
650 XX=X:YY=Y
660 GOTO 430
670 REM ********************
680 REM *** NO LIVES LEFT ***
690 REM ********************
700 FOR V=0 TO 3:SOUND V,0,0,0:NEXT V
710 I=1
720 POKE 87,1:GG=5:M$=""sorry but you
have lost all your lives"
  ":GOSUB 1300
730 IF SC>HI THEN GG=5:M$=""you have a
ttained a new high score"
  ":GOSUB 1300
740 IF SC<HI THEN GG=5:M$=""you did no
t manage to reach a new high score"
            ":GOSUB 1300
750 IF SC>HI THEN HI=SC
760 GG=5:M$=""press fire button for an
other go"              ":V=1:GOSUB 13
00
770 IF STRIG(0)=1 THEN GOTO 760
780 LV=3:FLAG1=0:FLAG2=0:MS=12:FL=500:
SC=0:LND=0:POP :GOTO 130
790 REM *************************
800 REM *** INITIAL SAUCER POS. ***
810 REM *************************
820 X=INT(RND(0)*17)+1
830 FOR K=23 TO 10 STEP -1
840 LOCATE X,K,LL:IF LL<>32 THEN POSIT
ION X,K-2:? #6;"◢":Y=K-2:POSITION X,K-
1:? #6;CHR$(ASC("/")-32):RETURN
850 NEXT K:RETURN
860 REM ************************
870 REM ***CHECK FOR SAUCER HIT***
880 REM ************************
890 IF SPOS=8 THEN GOTO 600
```

continued overleaf

LANDSCAPE continued from page 21

```
900 IF SPOS=36 OR SPOS=37 OR SPOS=38 O
R SPOS=39 OR SPOS=164 THEN X=XX:Y=YY:C
=1:GOSUB 1040:GOTO 600
910 IF SPOS=15 THEN GOSUB 950:GOTO 600
920 C=1:HT=HT+1:GOSUB 1040:GOTO 600
930 GOTO 930
940 REM ********************
950 REM *** SAUCER ON BASE **
960 REM ********************
970 IF HT>=14 THEN GOSUB 1410
980 X=XX:Y=YY:RETURN
990 REM *************************
1000 REM **CLEAR BOTTOM OF SCREEN**
1010 REM *************************
1020 POKE 87,1:FOR W=6 TO 22:POSITION
0,W:? #6;"                    ";:NEXT
W:RETURN
1030 REM *******************
1040 REM *** LIVES UPDATE ***
1050 REM *******************
1060 FL=500:MS=12:POKE 87,1:POSITION X
,Y:? #6;"":FOR Q=0 TO 20:SOUND 0,25,0
,15:NEXT Q:SOUND 0,0,0,0:LV=LV-1
1070 POKE 77,0:SOUND 2,0,0,0:SOUND 1,0
,0,0
1080 MSL=30:FL=500
1090 IF LV=2 THEN FLAG1=0:FLAG2=0:GG=5
:M$="you have just lost a life
        ":GOSUB 1300
1100 IF LV=1 THEN FLAG1=0:FLAG2=0:GG=5
:M$="you have just lost another life
             ":GOSUB 1300
1110 IF LV=1 THEN M$="you are now on
your final life                 ":GOSU
B 1300
1120 IF LV=0 THEN 1170
1130 IF HT>=14 THEN GOSUB 1200
1140 IF C=1 THEN C=0:GOSUB 990:GOSUB 1
710:GOSUB 800:POSITION 3,5:? #6;"
":LND=0:GOSUB 1200:GOTO 1160
1150 GOSUB 990:GOSUB 1710:POSITION 3,5
:? #6;"":LND=0:GOSUB 120
0
1160 IF LV=2 THEN POKE 87,0:POSITION 1
,1:? "Lives: ( (  ";:POKE 87,1:RETURN
1170 IF LV=1 THEN POKE 87,0:POSITION 1
,1:? "Lives: (    ";:POKE 87,1:RETURN
1180 IF LV<=0 THEN POKE 87,0:POSITION
1,1:? "Lives: ZERO ";:POP :GOTO 680
1190 REM *******************
1200 REM *** SCORE UPDATE ***
1210 REM *******************
1220 A$="00000":A$(6-LEN(STR$(SC)))=ST
R$(SC)
1230 B$="000":B$(4-LEN(STR$(FL)))=STR$
(FL)
1240 POKE 87,0
1250 POSITION 21,0:? A$:POSITION 36,0:
? MS;" ":POSITION 6,0:? B$
1260 POSITION 30,1:? #6;"LANDED:";LND:
IF LND>=5 THEN LND=0:C=1:GOSUB 1040:RE
TURN
1270 POKE 87,1:RETURN
1280 RETURN
1290 GOTO 1290
1300 REM ************************
1310 REM *** SCROLL BOTTOM LINE ***
1320 REM ************************
1330 FOR L=1 TO LEN(M$)
1340 IF 17-L<2 THEN F=F+1
1350 POSITION 17-L+F,23:? #6;M$(1+F,L)
1360 FOR G=0 TO GG:NEXT G
1370 NEXT L:F=0:RETURN
1380 REM *************************
1390 REM **SAUCER ON BASE MESSAGE**
1400 REM *************************
1410 FL=500:MS=12:FOR I=0 TO 3:SOUND I
,0,0,0:NEXT I:GOSUB 1200
1420 IF HT>=14 THEN GG=3:M$="well don
e! you are going up a level
     ":GOSUB 1300
1430 IF HT>=14 THEN IF DIF<3 THEN DIF=
DIF+1
1440 IF HT>=14 AND DIF=2 THEN GG=3:M$=
"you are now on skill level two
          ":GOSUB 1300
1450 IF HT>=14 AND DIF=3 THEN GG=3:M$=
"you are now on skill level three
             ":GOSUB 1300
1460 IF HT>=14 THEN HT=0:GOSUB 990:MSL
=30:FL=500:LND=0:POP :GOSUB 1710:GOTO
330
1470 REM **SAUCER MISSILE MOVEMENT**
1480 MY=MY-1:IF MY<=5 THEN FLAG1=0:POS
ITION MX,MY+1:? #6;" ":SOUND 2,0,0,0:G
OSUB 1200:RETURN
1490 SOUND 2,25-MY,8,15
1500 LOCATE MX,MY,MPOS:IF MPOS=32 THEN
 1530
1510 GOSUB 1560:IF MPOS=ASC(",") THEN
MS=12:FL=500:SC=SC+500:G=19
1520 IF MPOS=ASC("-")-32 THEN SC=SC+10
0:HT=HT+1:FLAG2=0:FLAG1=0:SOUND 2,0,0,
0:GOSUB 1200
1530 POSITION MX,MY:? #6;CHR$(9):POSIT
ION MX,MY+1:? #6;" "
1540 IF FLAG1=0 THEN POSITION MX,MY:?
#6;" "
1550 RETURN
1560 POSITION MX,MY:? #6;CHR$(ASC("+")
-32)
1570 FOR H=1 TO 20:SOUND 2,46-H,0,15:N
EXT H:POSITION MX,MY:? #6;" ":RETURN
1580 POSITION X,Y:? #6;CHR$(ASC("+")-3
2)
1590 FOR H=1 TO 20:SOUND 2,46-H,0,15:N
EXT H:RETURN
1600 REM ********************
1610 REM ***ALIEN MOVEMENT ***
1620 REM ********************
1630 AAY=AY:AY=AY+1
1640 LOCATE AX,AY,APOS
1650 IF APOS=36 OR APOS=37 OR APOS=38
OR APOS=39 OR APOS=15 THEN POSITION AX
,AY-1:? #6;" ":FLAG2=0:F=1:HT=HT+1
1660 IF F=1 THEN F=0:LND=LND+1:SOUND 0
,50,8,15:POSITION AX,AY-1:? #6;"+":GOS
UB 1200:SOUND 0,0,0,0:POSITION AX,AY-1
:? #6;" ":RETURN
1670 IF APOS=ASC(")")-32 THEN FLAG1=0:
FLAG2=0:POSITION AX,AY:? #6;" ":MY=MY-
1:GOSUB 1560:SOUND 2,0,0,0:HT=HT+1
1680 IF APOS=ASC(")")-32 THEN GOSUB 12
00:RETURN
1690 POSITION AX,AY:? #6;"¯":POSITION
AX,AAY:? #6;" ":RETURN
1700 REM ****************
1710 REM *** LANDSCAPE ***
1720 REM ****************
1730 HT=0
1740. Y=INT(RND(0)*6)+13:K=-1:YY=20
1750 FOR D=0 TO 19
1760 IF YY=Y OR YY>=22 OR YY<=7+((4-DI
F)*3) THEN K=-K:Y=INT(RND(0)*6)+13:D=D
-1
1770 YY=YY+K
1780 IF K=-1 THEN POSITION D,YY:? #6;C
HR$(INT(RND(0)*2)+36):POSITION D,YY+1:
? #6;"":POSITION D,YY-1:? #6;" "
1790 IF K=1 THEN POSITION D,YY:? #6;CH
R$(INT(RND(0)*2)+38):POSITION D,YY+1:?
 #6;"":POSITION D,YY-1:? #6;" "
1800 NEXT D:RETURN
1810 REM *************************
1820 REM ***CHARACTER DEFINITION***
1830 REM *************************
1850 P=PEEK(106)+1
1860 N=P*256
1870 L=1023:FOR M=0 TO 1023:L=L-1:POSI
TION 9,5:? #6;INT(L/30);" ":POKE N+M,P
EEK(57344+M):NEXT M
1880 POSITION 9,5:? #6;"  ":POSITION 4
,4:? #6;" energizing ":FOR J=4 TO 15
1890 FOR M=0 TO 7:READ D:POKE N+M+(J*8
),D:NEXT M:NEXT J
1900 POKE 756,P
1910 RETURN
1920 REM *************************
1930 REM **DATA FOR NEW CHARACTERS**
1940 REM *************************
```

continued on page 39

SCREENDUMP

Computer Art comes to PAGE 6! The following screens were done by Harvey Kong Tin of New Zealand using MicroPainter. The screens were dumped on an NEC 8023 printer using Megafont.







If you have any pictures suitable for **SCREENDUMP** *please send them in. The screens can be drawn with any utility but must be saved in MicroPainter compatible format. Please send files on disk with a brief note of how they were drawn and some details of yourself to the editorial address on page 3. All disks will be returned.*

Programming

What is USR?

USR is probably the least well-documented function in ATARI BASIC, yet it is potentially one of the most powerful. This introduction is in three parts. First we look at what USR is and the syntax it uses. This is for BASIC programmers who have come across it in program listings and just want a feel for what it does. The second part looks at the way USR works and outlines the general principles of inserting machine-code subroutines into BASIC. If you are not into machine-code and don't intend to start writing your own routines, you can comfortably skip this bit. Finally, we look at some examples you can experiment with, including a routine for copying the ROM character set into RAM at lightning speed.

Let's start with the simplest form a USR statement can take. This looks something like X=USR(1536). In English, this means 'Stop executing BASIC for a moment, go execute the machine-code routines which start at address 1536 and put the resulting number into variable X'. In the majority of cases you won't give two hoots what the value of X turns out to be, what is important is the execution of the subroutine along the way. If it is designed to move a player about on the screen, your only concern is whether the movement works. If it is a scrolling routine, then it is the actual scrolling that counts. Any hypothetical number generated at the end of the routine would probably have little relevance to the real world anyway and you would hardly ever use it for anything. So why bother assigning a variable to it?

The answer is that USR is not a command like GOSUB or POKE. It is a function, like PEEK or INT, so we can't simply type USR(1536), we have to give it a command to work with. Theoretically, we could use any command that works with a number. TRAP or RESTORE would do, provided we knew that the number would never exceed 32767 (the maximum allowed with these commands). PRINT would also work but would mess up your screen display. The most convenient command is LET using the format 'LET X=' or, more simply 'X=', since this does not place restrictions on the number following it and has no discernable effect on program execution.

You are not stuck with X as a variable name of course. Some programmers prefer statements like MOVE=USR(1536) or SCROLL=USR(1536) to give a clue about the subroutine's purpose. Similarly, the number in brackets need not be 1536 although that is a common one since it is the first location of an area (page 6) specially reserved for things like machine-code subroutines. It need not even be a 'raw' number. Variable names or expressions (like Z*3+5) are equally acceptable, provided they evaluate to the correct starting address. The thing to remember is that, however many numbers appear in the brackets, the first one is *always* the place where the machine-code routine starts.

Now a machine-code program is simply a long list of numbers, each representing either a command or an item of data, depending on its position. You can, if you like, see what the routine looks like by finding the place in your BASIC listing where the numbers are put into memory, starting at the address given in the USR statement. There are numerous ways of getting a list of numbers into RAM. Very large routines might be loaded from cassette or disk directly into the chosen memory area, using something known as a 'direct CIO call', but it is unlikely you will encounter this method in public domain BASIC programs. A very common technique is to POKE the numbers one at a time into RAM, using READ and DATA statements. A third approach is to DIMension a string to the length of the machine code routine, then store the list of numbers in that string as ATASCII symbols. In such cases, the USR statement will take the form of X=USR(ADR(A$)) and you will find A$ written out somewhere in the listing, looking like a meaningless jumble of characters and symbols. This method saves both space and time, since it eliminates the need for a machine-code loading program, but it is extremely vulnerable to typing errors and a single mistake can crash the system. One final, and little used, approach is to put very short machine code routines into the USR statement itself, either as extra numbers or ATASCII symbols. De Re Atari gives the example X=USR(ADR("hhh/*/LV/d"),16) but I have never seen it used anywhere else and most subroutines will be too long to encode in this fashion. My personal preference is for the POKE, READ and DATA technique. This is much kinder for anyone who has to copy the program from a listing and makes debugging a lot easier.

Let's now look at the other numbers you might find in a USR statement's brackets. How about X=USR(1536, 100, 20, 3000, PLR1, MEMTOP-10)? These extra numbers are known as 'parameters' to make the point that they are not

Len Golding explains....

addresses. They are just ordinary numbers which will be used somewhere in the machine-code routine called by that USR statement. A parameter can be a real number, a variable name or an expression, the only restriction is that it must evaluate to a number between 0 and 65536. It might indicate which joystick the routine should read, or which player to move, or which colour register to use, or where in memory to find some data, or where to store a result - almost anything in fact. It is even possible, by using variable names, to pass the result of an earlier calculation carried out in BASIC, like how much memory there is left at any given time, or where to put an explosion on the screen. A USR statement can contain up to 255 parameters, but you are not likely to encounter more than half a dozen or so.

Discovering what the parameters mean in any given instance is a thankless task unless the programmer has deliberately made it easy for you. Sometimes he or she will have used variable names whose function can be identified from a close inspection of the BASIC listing. Alternatively, there may be a REM statement close by explaining all. If neither of these applies, there is normally no easy way of discovering what the parameters mean, or how the machine code routine uses them. Just type them in and trust the programmer!

Now on to the second part, how USR works. I'll assume that if you are reading this section you know about converting decimal numbers to 2-byte integers and how a LIFO stack operates. If not, skip the next couple of paragraphs. Better still, get hold of a decent book on machine code and find out!

When a USR call is made, the following things happen:

a) The processor notes where it is in the BASIC program, and pushes this location onto the stack for later use as a return address.
b) Any parameters passed are converted into 2-byte integers and pushed onto the stack, low byte first.
c) A one-byte value containing the number of parameters passed (even if it is 0) is pushed onto the stack.
d) the machine code routine is executed.
e) On encountering the final RTS instruction, the top two bytes are pulled off the stack and used as the return address. All being well, this transfers control back to BASIC, at the next statement after the USR call.

Note that I say 'all being well'. A number of things can go wrong if we're not careful. First of all, there is that byte mentioned at c), sitting on top of the stack ready to foul things up. Unless we get rid of it, the processor will think it is part of the return address and, when the final RTS is encountered, will bounce off into the lower reaches of operating system RAM instead of returning to BASIC. Consequently it is a good idea to do a PLA right at the start of your routine to be sure you don't forget it. For exactly the same reason, all parameters have to be pulled off the stack before the final RTS. They can be left there until you need them in your routine of course, but newcomers to machine-code programming may find it safer to retrieve them at the start of the routine. They can always be stored in a less critical location until you need them.

Remember that all parameters are converted into 2-byte integers, even if their value could be contained in a single byte, so to retrieve a one-byte parameter, you have to do two PLAs and discard the high byte. Also, don't forget that parameters come back off the stack with their high byte first. This can be a bit confusing if you are used to the conventional 'low/high' order of storing 2-byte numbers. Lastly, do make sure that your routine ends with an RTS. I know this sounds obvious, but it is easy to forget, especially if the routine uses a lot of JSRs (e.g. accessing ROM routines).

On now to the third section, where we get down to some practical examples. Here is the simplest machine code routine I can think of

```
PLA          ;Get rid of the number of
              parameters byte
LDA 20
STA 710      ;Store 20 in the address
              controlling screen
              colour
RTS          ;Return to BASIC
```

In decimal form, this routine translates to : 104,169,20,141,198,2,96. Before we can do a USR call though, these numbers have to be put into some safe area of memory. Let's use address

continued overleaf

USR continued

1536 onwards. Here is my favourite way of loading machine code subroutines into RAM, though it is not necessarily the best.

```
10 X=0:RESTORE 40
20 READ D:IF D=-1 THEN 100
30 POKE 1536+X,D:X=X+1:GOTO 20
40 DATA 104,169,20,141,198,2,96,-1
100 X=USR(1536)
```

As with any machine-code routine, SAVE it before you RUN it, since the slightest error in the DATA line could lock up your system. When you RUN it, you should find that the screen turns orange. Okay, it's a trivial example, you could have achieved the same effect by POKE 710,20, but at least it's a start. Notice the 96 at the end of the routine. This is the RTS instruction which we need to get back into BASIC.

How about an example with a parameter? This slightly modified routine allows you to specify the screen colour within the USR statement.

```
PLA        ;Discard the number of
            parameters byte
PLA        ;Discard the parameters
            high byte
PLA        ;Get the parameters
            low byte
STA 710    ;Store it in the colour
            register
RTS        ;Return to BASIC
```

Subroutine writers note: even though this particular parameter can only be a one-byte number (you can't put more than 255 into a colour register), the USR call will still push two bytes onto the stack, so we have to pull the high byte off and throw it away. To load this more flexible subroutine, change lines 40 and 100 of the BASIC loader program above to read:

```
40 DATA 104,104,104,141,198,2,96,-1
100 X=USR(1536,90)
```

This time the screen should turn red, but you can choose whatever colour you like, simply by altering the 50 in the brackets.

Finally (and at long last) something useful. Anyone who has used redefined character sets in their programs knows that, before any redefinition can be done, the character set has to be copied from ROM into RAM. This takes about 10 to 15 seconds in BASIC, depending on the method used. The following machine-code routine does it in the blink of an eye! Alter the BASIC loader program as follows

```
40 DATA 104,104,133,204,104,133,203,16
9,224
50 DATA 133,206,160,0,132,205,162,4
60 DATA 177,205,145,203,136,208,249,23
0
70 DATA 204,230,206,202,208,242,96,-1
100 X=USR(1536,10240)
```

The parameter 10240 is, in this case, the address where the copied character set will start. It need not be 10240 of course, you might prefer to calculate where the top of your useable memory is and put the new character set up there, remembering to leave enough space for the screen memory and display list and 1024 bytes for the characters. Try changing line 100 to read

```
100 START=256*(PEEK(106)-20):X=USR(153
6,START)
```

When you run it for the first time, the routine does not look particularly fast. This is because of the time taken to load the numbers into memory, but, once there, the routine can be called as many times as you like and will execute very quickly indeed. To see it at full speed, RUN it once then type GOTO 100. The real benefits arrive, of course, when you use the routine more than once. You can copy the character set three or four times over (to different locations of course) in a couple of seconds. This is useful if you want several incarnations of your redefined characters to get animation effects of the kinds used in Space Invaders. That subject, however, deserves another article all of its own.

In conclusion, let's look at some of the reasons for going to all this bother. There are two main ones. Firstly machine-code can do some things BASIC can't manage at all, and secondly, it executes up to 200 times faster. On the other hand, machine-code takes longer to write, is far more exacting and makes complicated tasks out of some things BASIC finds easy. The USR function gives us the best of both worlds. We can use BASIC to do all the things which would be very tedious in machine-code, like complex arithmetic, or setting up screen displays, or using peripherals and employ machine-code for tasks which need to be executed at high speed like moving player/missiles vertically, or reading light pens, or fine scrolling.

All in all, it's well worth making friends with USR. ●

Education

FLAGS

by Keith Berry

The computer is an ideal tool for education but one of the problems is finding the right context in which to use it. All too often programs of an educational nature use straight text for revision and accomplish little more than can be learned from a book.

FLAGS uses the best features of the Atari - colour and sound - to present a Quiz or Display of the National flags of many countries. By running through the Display option your children (and you!) can learn the various national flags and follow this up with a quiz. Flags which are not correctly identified are summarised at the end for further revision.

```
0 REM ********************************
1 REM **            FLAGS           **
2 REM **      by Keith Berry,       **
3 REM **    50, Brantley Road,      **
4 REM **     Birmingham B6 7DR      **
5 REM ********************************
10 REM
15 READ RESET,LOOP,CSET,FIN,CLEAR,YES,NO,OS,MULTI,ECU,VERT,TITLE
16 DATA 130,140,160,840,950,990,1040,1100,1170,1280,1320,1880
25 READ SCAN,SWIS,HORIZ,FILL,QTR,WS,CNR,MC,DPLY,FANF,ADP
26 DATA 1360,1470,1520,1560,1590,1630,1650,1690,1720,1800,1920
35 READ C0,C1,C2,C3,C5,C6,C10,C12,C13,C15,C18,C17
36 DATA 0,1,2,3,5,6,10,12,13,15,18,17
45 READ C19,C35,C255,INK,CR1,CR2,CR3,CR5,CLS,MAR
46 DATA 19,35,255,764,708,709,710,712,125,82
60 DIM ALT$(16),G$(C1),FLAG$(C12),AN$(16),NAME$(8),MISS$(390),TEMP$(C13),DA(30)
65 GOSUB 2000
70 GOSUB TITLE
90 ? :? " national flags   DISPLAY OR QUIZ?"
100 FOR W=C1 TO 2000:Z=PEEK(INK):IF Z=47 THEN GOTO 120
110 IF Z=58 THEN POKE INK,C255:GOSUB CLEAR:RESTORE 400:GOTO DPLY
115 NEXT W:Z=58:GOSUB ADP:GOSUB CLEAR:RESTORE 400:GOTO DPLY
120 POKE INK,C255:? CHR$(CLS);"  welcome to flags    PLEASE ENTER YOUR   NAME: ";:INPUT NAME$
125 IF LEN(NAME$)<C2 THEN 120
130 M1=C1:I=C0:SC=C0:Q=C0:T=150:FOR W=C1 TO 30:DA(W)=C0:NEXT W
140 ? CHR$(CLS):GOSUB CLEAR
142 I=I+C1
144 RS=INT(RND(0)*45)+C1
146 FOR J=C1 TO I:IF DA(J)=RS THEN POP:GOTO 144
148 NEXT J:DA(I)=RS
150 RESTORE RS+400:READ ALT$,FLAG$,G,P1,P2,P3,P4
160 POKE CR1,P1:POKE CR2,P2:POKE CR3,P3:POKE CR5,P4
170 ON G GOSUB MC,ECU,VERT,QTR,HORIZ,CNR,SCAN,WS,MULTI,SWIS,OS
280 IF Z=58 THEN RETURN
290 ? "  THIS IS THE FLAG":? "   OF WHICH COUNTRY";:TRAP 290:INPUT AN$
300 TRAP 40000
310 ? CHR$(CLS)
320 IF AN$=ALT$ OR AN$=FLAG$ THEN ? "  ";FLAG$;" IS":? "  CORRECT, ";NAME$;"!":? :GOSUB YES:SC=SC+C1
330 IF AN$<>ALT$ AND AN$<>FLAG$ THEN MIS=MIS+C1:GOSUB 380
340 POKE INK,C255:FOR W=C1 TO 400:IF PEEK(INK)=C12 THEN 360
350 NEXT W
360 POKE INK,C255:IF I=30 THEN GOTO FIN
370 GOTO LOOP
380 ? "  NO, ";NAME$;".":? "  IT'S ";FLAG$:? :GOSUB NO
385 TEMP$=FLAG$:LN=LEN(FLAG$):IF LN<C13 THEN FOR Z=LN+C1 TO C13:TEMP$(Z)=" ":NEXT Z
390 M2=M1+C13:MISS$(M1,M2)=TEMP$:M1=M1+C13
395 RETURN
401 DATA SIERRA LEONE,SIERRA LEONE,5,212,12,118,26
402 DATA SALVADOR,EL SALVADOR,5,148,12,148,24
403 DATA CHAD,CHAD,3,114,26,52,120
404 DATA HAITI,HAITI,4,52,0,0,120
405 DATA SIAM,THAILAND,9,52,12,114,26
406 DATA ANDORRA,ANDORRA,5,112,24,52,120
407 DATA VOLTAIC REPUBLIC,UPPER VOLTA,5,0,12,52,120
408 DATA ARGENTINA,ARGENTINA,5,118,12,118,0
409 DATA NORWAY,NORWAY,7,52,12,112,202
410 DATA BOLIVIA,BOLIVIA,5,52,24,196,8
411 DATA FRANCE,FRANCE,3,114,12,52,120
412 DATA WESTERN SAMOA,W.SAMOA,8,52,52,114,12
413 DATA HOLLAND,NETHERLANDS,5,52,12,112,120
414 DATA MALTA G.C.,MALTA,1,52,12,12,24
415 DATA MADAGASCAR,MALAGASY,11,12,52,196,6
416 DATA ALAND ISLANDS,AALAND,7,130,24,52,186
417 DATA EIRE,IRELAND,3,196,12,24,120
418 DATA HELVETIA,SWITZERLAND,10,0,52,12,118
419 DATA AUSTRIA,AUSTRIA,5,52,12,52,120
420 DATA ABU DHABI,ABU DHABI,4,52,52,12,120
421 DATA ITALIA,ITALY,3,196,12,52,120
422 DATA GUATEMALA,GUATEMALA,3,148,12,148,24
423 DATA SWEDEN,SWEDEN,7,118,118,26,110
424 DATA EQUADOR,ECUADOR,2,24,148,52,210
425 DATA WESTERN GERMANY,WEST GERMANY,5,0,52,248,120
426 DATA TONGA ISLAND,TONGA,6,52,52,12,120
427 DATA SOUDAN,MALI,3,196,26,52,10
428 DATA HUNGARY,HUNGARY,5,52,12,198,24
429 DATA AJMAN,AJMAN,4,52,12,12,120
430 DATA DANMARK,DENMARK,7,52,52,12,120
431 DATA PERSIA,IRAN,5,196,12,52,120
432 DATA GUINEA,GUINEA,3,52,26,196,10
433 DATA TRANSKEI,TRANSKEI,5,50,12,196,26
434 DATA COSTARICA,COSTA RICA,9,116,12,52,26
435 DATA FAROE ISLANDS,FAROE,7,12,132,52,198
436 DATA LUXEMBOURG,LUXEMBURG,5,52,12,116,120
437 DATA IVORY COAST,IVORY COAST,3,24,12,196,114
438 DATA SUOMI,FINLAND,7,12,118,118,22
```

▶▶

FLAGS continued from page 29

```
439 DATA GABON,GABON,5,196,24,114,118
440 DATA PERU,PERU,3,52,12,52,136
441 DATA NIGERIA,NIGERIA,3,196,12,196,
24
442 DATA BULGARIA,BULGARIA,5,12,196,52
,134
443 DATA BENIN,DAHOMEY,11,196,26,52,12
0
444 DATA BELGIUM,BELGIUM,3,0,26,52,10
445 DATA ETHIOPIA,ETHIOPIA,5,196,24,52
,120
446 DATA ICELAND,ICELAND,7,114,12,52,1
0
447 DATA Z,END,0,0,0,0,0
840 GRAPHICS C18:POKE CR1,C12:POKE CR3
,8:POKE 711,118:POKE CR5,52:? #C6;"
"
850 ? #C6
860 ? #C6;"AT THE END OF THE"
870 ? #C6;"TEST, ";NAME$;", YOU"
880 ? #C6;"HAVE SCORED ";SC;" OUT"
890 ? #C6;"OF ";I;"."
900 ? #C6
910 ? #C6;"PRESS RETURN TO          CONT
INUE"
912 Z=PEEK(INK):IF Z<>C12 THEN 912
920 GRAPHICS C0:POKE MAR,C1:POKE 752,C
1:? :? :? :? "         *** NATIONAL FLAG
S ***"
925 POKE CR3,C12:POKE CR2,C0:POSITION
C1,C6:? " YOU MIS-IDENTIFIED THE FOLLO
WING: "
930 ? :? :? MISS$:? :? "LOOK FOR THEM
IN DISPLAY MODE.":POKE INK,C255:? :? :
? "Press  RETURN  to RUN again:"
935 Z=PEEK(INK):IF Z<>C12 THEN 935
940 RUN
950 POKE MAR,C0:GRAPHICS C3:POKE CR3,C
0:A=PEEK(560)+PEEK(561)*256
960 IF PEEK(A)<>66 THEN A=A+C1:GOTO 96
0
970 POKE A,70:POKE A+C3,C6:POKE A+4,C6
:POKE A+C5,C6
980 RETURN
990 FOR N=C1 TO C3
1000 FOR L=C0 TO C15:SOUND C0,60,C10,L
:SOUND C1,40,C10,L:NEXT L
1010 FOR L=C15 TO C0 STEP -C1:SOUND C0
,50,C10,L:SOUND C1,C10,C10,L:NEXT L
1020 NEXT N
1030 RETURN
1040 FOR N=C1 TO C3
1050 FOR L=C15 TO C0 STEP -0.25
1060 SOUND C1,200,C12,L
1070 NEXT L
1080 NEXT N
1090 RETURN
1100 COLOR C1:FOR P=C1 TO C18
1110 PLOT C5,P:DRAWTO C15,P:NEXT P
1120 COLOR C2:FOR P=C1 TO 9
1130 PLOT 16,P:DRAWTO C35,P:NEXT P
1140 COLOR C3:FOR P=C10 TO C18
1150 PLOT 16,P:DRAWTO C35,P:NEXT P
1160 RETURN
1170 COLOR C1:FOR P=C1 TO C3
1180 PLOT C5,P:DRAWTO C35,P:NEXT P
1190 COLOR C2:FOR P=4 TO C6
1200 PLOT C5,P:DRAWTO C35,P:NEXT P
1210 COLOR C3:FOR P=7 TO C12
1220 PLOT C5,P:DRAWTO C35,P:NEXT P
1230 COLOR C2:FOR P=C13 TO C15
1240 PLOT C5,P:DRAWTO C35,P:NEXT P
1250 COLOR C1:FOR P=16 TO C18
1260 PLOT C5,P:DRAWTO C35,P:NEXT P
1270 RETURN
1280 COLOR C1:FOR P=C1 TO 9:PLOT C5,P:
DRAWTO C35,P:NEXT P
1290 COLOR C2:FOR P=C10 TO 14:PLOT C5,
P:DRAWTO C35,P:NEXT P
1300 COLOR C3:FOR P=C15 TO 20:PLOT C5,
P:DRAWTO C35,P:NEXT P
1310 RETURN
1320 C=C1:COLOR C:PLOT C15,C17:DRAWTO
C15,C0:DRAWTO C5,C0:POSITION C5,C17:GO
SUB FILL
1330 C=C2:COLOR C:PLOT 25,C17:DRAWTO 2
5,C0:DRAWTO C15,C0:POSITION C15,C17:GO
SUB FILL
1340 C=C3:COLOR C:PLOT C35,C17:DRAWTO
C35,C0:DRAWTO 25,C0:POSITION 25,C17:GO
SUB FILL
1350 RETURN
1360 COLOR C1:FOR P=C0 TO C17
1370 PLOT C5,P:DRAWTO C35,P:NEXT P
1380 COLOR C2:FOR P=7 TO 11
1390 PLOT C5,P:DRAWTO C35,P:NEXT P
1400 COLOR C2:FOR P=C12 TO 16
1410 PLOT P,C0:DRAWTO P,C17:NEXT P
1420 COLOR C3:FOR P=C13 TO C15
1430 PLOT P,C0:DRAWTO P,C17:NEXT P
1440 COLOR C3:FOR P=8 TO C10
1450 PLOT C5,P:DRAWTO C35,P:NEXT P
1460 RETURN
1470 COLOR C2:FOR P=C0 TO C18:PLOT C10
,P:DRAWTO 30,P:NEXT P
1480 COLOR C3:FOR P=C2 TO 16:PLOT C19,
P:DRAWTO 21,P:NEXT P
1490 COLOR C3:FOR P=8 TO C10:PLOT C12,
P:DRAWTO 28,P:NEXT P
1500 RETURN
1510 END
1520 COLOR C1:FOR P=C0 TO C5:PLOT C3,P
:DRAWTO C35,P:NEXT P
1530 COLOR C2:FOR P=C6 TO 11:PLOT C3,P
:DRAWTO C35,P:NEXT P
1540 COLOR C3:FOR P=C12 TO C17:PLOT C3
,P:DRAWTO C35,P:NEXT P
1550 RETURN
1560 POKE 765,C
1570 XIO C18,#C6,C0,0,"S:"
1580 RETURN
1590 C=C1:COLOR C:PLOT C35,C18:DRAWTO
C35,C0:DRAWTO 20,C0:POSITION 20,C18:GO
SUB FILL
1600 C=C2:COLOR C2:PLOT C19,C18:DRAWTO
 C19,9:DRAWTO C5,9:POSITION C5,C18:GOS
UB FILL
1610 C=C3:COLOR C:PLOT C19,8:DRAWTO C1
9,0:DRAWTO C5,C0:POSITION C5,8:GOSUB F
ILL
1620 RETURN
1630 GOSUB QTR
1640 COLOR C0:PLOT C12,C2:PLOT C12,C6:
PLOT 8,C3:PLOT C17,4:PLOT 14,C5:RETURN
1650 GOSUB QTR
1660 COLOR C1:FOR P=C3 TO C5:PLOT 8,P:
DRAWTO 16,P:NEXT P
1670 FOR P=11 TO C13:PLOT P,C1:DRAWTO
P,7:NEXT P
1680 RETURN
1690 GOSUB QTR
1700 COLOR C0:PLOT C6,C2:DRAWTO 8,C2
1710 PLOT 7,C1:DRAWTO 7,C3:RETURN
1720 READ ALT$,FLAG$,G,P1,P2,P3,P4
1730 IF FLAG$="END" THEN RESTORE :GOTO
 70
1740 GOSUB CSET
1750 ? CHR$(CLS):FOR S=C1 TO C10-(LEN(
FLAG$)/C2):? " ";:NEXT S:? FLAG$:GOSUB
 FANF
1760 POKE INK,C255:FOR W=C1 TO 400:IF
PEEK(INK)=C12 THEN 1780
1770 NEXT W
1780 POKE INK,C255:GOSUB CLEAR
1790 POKE 77,C0:GOTO DPLY
1800 FOR S=C1 TO C3
1810 FOR V=0 TO C12:SOUND C0,150,C10,V
:SOUND C1,151,C10,V:SOUND C2,152,C10,V
:NEXT V
1820 FOR V=C15 TO C0 STEP -C5:SOUND C0
,150,C10,V:SOUND C1,151,C10,V:SOUND C2
,152,C10,V:NEXT V
1830 NEXT S
1840 T=T-C10:IF T<50 THEN T=150
1850 FOR V=C0 TO C15:SOUND C0,T,C10,V:
SOUND C1,T+C1,C10,V:SOUND C2,T+C2,C10,
V:NEXT V:FOR W=C1 TO 50:NEXT W
1860 FOR V=C15 TO C0 STEP -0.25:SOUND
C0,T,C10,V:SOUND C1,T+C1,C10,V:SOUND C
2,T+C2,C10,V:NEXT V
1870 RETURN
1880 POKE MAR,C0:GRAPHICS C3:POKE CR3,
C0:A=PEEK(560)+PEEK(561)*256
1890 IF PEEK(A)<>66 THEN A=A+C1:GOTO 1
890
```

continued on page 39

CONTACT

SEIKOSHA GP-100A PRINTER: Can anyone tell me how to get graphics or a simple screen dump? Neil Scott, 110, Coventry Road, Bulkington, Nr. Nuneaton, Warks, CV12 9NJ. Tel. 0203 316421

CORBY USERS GROUP: I am thinking of setting up a User Group in the Corby area. Anyone interested? Marco Dawson, 15, Tyne Road, Corby, Northants, NN17 2HU

ASSEMBLER PROGRAMMING: Anyone in my area interested in swapping ideas etc? Meet or write. Steve Hill, 5 Broadacres, Durkar, Wakefield, W.Yorks, WF4 3BE

JOURNEYMAN: I travel regularly throughout Yorks, Lancs, Cheshire, Notts and Derbyshire and would like to meet other addicts. Give me a ring. Alan Goldsbro, 51 Inglewood Place, Leeds, LS14 6HH. Tel. Leeds 600754

OXFORD USERS GROUP: Any support for a User Group in Oxford? Please contact Matthew Spencer with any ideas etc. Tel. Stonesfield 757 or write 16, Pumbro, Stonesfield, Oxford, OX7 2QF

ANALOG: Back issues 1 to 7 required urgently. Tel Mr L. Lazarus 01-202 8331

LIGHT PEN: Do you have a light pen program in Basic that will draw in Graphics 8? Please contact Jonathon Russell, 269 Campkin Road, Cambridge, CB4 2LE. Tel.0223 62006

NORTHERN IRELAND GROUP: Feeling left out? Anyone interested in a User Group in the Belfast area with the possibility of publishing a book and a software library? Please write with s.a.e. to Frankie Smyth, 62, Orchardville Ave., Belfast, BT10 0JH

GHOST TOWN/THE COUNT: I am completely baffled about how to shoe the horse in Ghost Town or get past the coffin in The Count. Can you help? Anthony Pottle, 22, Greenacres Drive, Lutterworth, Leics, LE17 4TG

PAGE 6 BACK ISSUES: The following readers require copies of issues 1 & 2

Derek Ross, 117 Moss-side Road, Shawlands, Glasgow G41. Tel. 041 632 5737

Martin Sisson, 45, Westerfield Way, Silverdale, Nottingham, NG11 7ET

ZORK 1: I am just starting the perilous journey and would welcome exchanging ideas or information in writing with another on the same journey. John Dimmer, 71 Duncan Drive, Elgin, Moray. Tel. 0343 44695

850 INTERFACE: Anyone got one to sell? John Dimmer (address as above).

THE SOFTWARE REVIEWS

***** STAR GAME *****

ATTACK OF THE MUTANT CAMELS

LLAMASOFT 16K 1/2 PLAYERS

In the words of the hairy Llama man himself. Like, WOW, man....this is a real blast! Mutant Camels comes to the Atari with a vengeance. Not just a translation but a full blown adaptation using all of the Atari's powerful features including display list interrupts, scrolling, player missile graphics, animation and AWESOME sound. Llamasoft have been bandying the word AWESOME about for some time but now that he has had the chance to pull the stops out of an Atari, I am sure that Jeff Minter now knows what AWESOME really means. You don't really need to read any further, just get your £7.50 (yes, that's all) off to Llamasoft and wait for a superbly packaged, excellently programmed, top quality cassette to drop through your door. You won't regret it.

For those of you who don't know anything of the Llama man's games, let me give you some details of this one. It helps to know a little about the game's designer, Jeff Minter, a kind of young eccentric in the grand English tradition who lives in a sixties time-warp, is in love with Llamas and writes wierd games about Mutant Camels and Sheep in Space. Don't let the eccentricity fool you, the man is a brilliant programmer and has his finger right on the pulse of the eighties micro/arcade scene. The game features an attempt by the race of Zzyax to stop mankind from leaving Earth and exploring the galaxy by using genetic engineering on camels to create 90 foot high, neutronium-shielded, laser spitting beasts designed to bring death and destruction to Earth. The beasts can be destroyed by neutron-canon bolts from a highly manouverable fighter ship and that is your task. The Camels march relentlessly across the screen and you must destroy them one by one by pumping them full of neutron bolts. Your ship is shielded but loses shield power each time you collide with a Camel or are hit by a laser bolt until nine hits destroy the ship. The colours are brilliant with display list interrupts or GTIA forming the sky, pyramids on the horizon and a forward scrolling foreground. If you destroy all Camels on a level you go into hyperwarp phase and must avoid missiles as the terrain scrolls by at incredible speed. Colours on the hyperwarp are even better.

The only game to have been reviewed in PAGE 6 which has been given the title STAR GAME! Partly because there has not been a better Atari game at this price, partly because it proves to the world that Atari is the best home computer around and last but not least because any man that is mad enough to admit publicly that he loves llamas must be alright. The fact that he can write brilliant programs helps too.

Atari software generally sells in quite small quantities compared with other micros and if Jeff Minter is to adapt his other games it will require you to show your dedication and support by buying this in your thousands. If you love arcade games, you could not spend £7.50 more effectively. Buy it, get all your friends round and show them what a great game on the best micro looks like and get them to buy both an Atari and Attack of the Mutant Camels.

Les Ellingham

* * * * * * * * * * *

GWENDOLYN 40K DISK

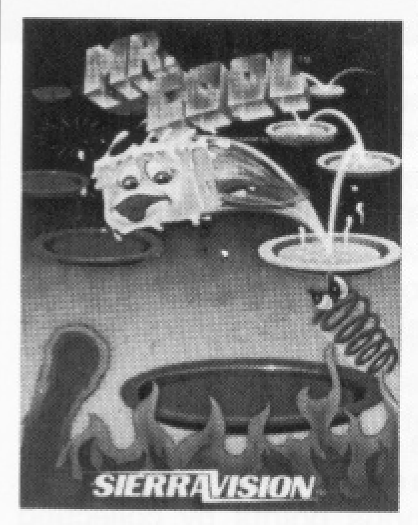
ARTWORX 1 PLAYER

Gwendolyn is an 'arcade-adventure' of the type which is fully joystick controlled and comes on two disks. The arcade adventure is much more graphically oriented than the text based adventures and Gwendolyn includes a certain amount of animation and sound. Whilst there are puzzles to solve the choices are limited and the plot is therefore not as complex as most text based adventures. The graphics are reasonably good but use artifacting which does not lend itself well to the PAL system and the images are therefore lacking in colour. At first I was quite disappointed with the game mainly because of the limitations of joystick control and I wandered around for a long time doing absolutely nothing, but once I was underground it became more interesting. Eventually I became hooked as more progress was made. One of the attractions of a two-disk adventure is to get onto the second disk! Gwendolyn is much more suited to novice or intermediate Adventurers or those who like some graphics and action. Experienced text Adventurers would probably find the joystick control too limiting and are likely to solve the adventure in a short time.

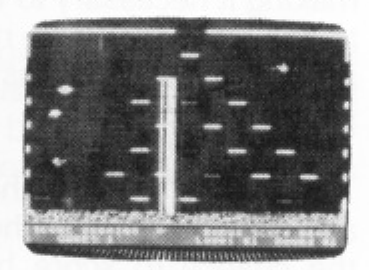
Les Ellingham

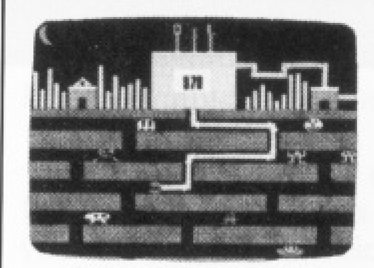
▶

ATARI® ZOOMSOFT SOFTWARE SPECIALIST

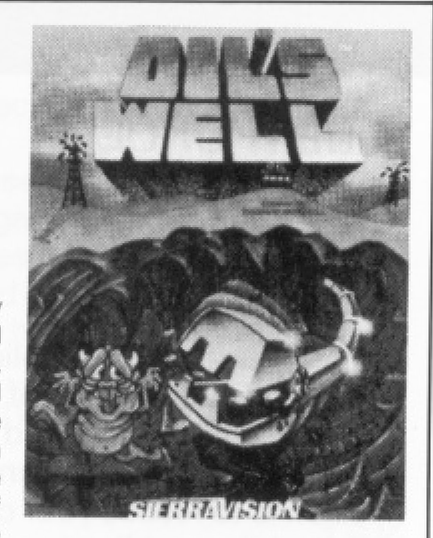


MR. COOL - Be ready for a new addiction that will be going around soon: "Mr. Cool-mania." This challenging new game will have people dodging fireballs and springs in their sleep. A fastmoving, smoothly animated game that will keep you busy for hours.





OILS WELL - An addictingly fun arcade game that will gobble its way to your heart. Drill underground to find oil deposits while avoiding the nasty creatures that can destroy your oil pipe. Multiple screens and changing levels of difficulty complete the



The PROTECT

The PROTECT allows you to write to both sides of any disk. Write protect without LABEL'S. No more cutting of notches that cause disk errors. Just plugs in. No soldering, takes only a few minutes to install.

Price £13.95

DISK COLLECTOR

DISK COLLECTOR is a complete disk cataloging system. 13 options to choose from. Stores over 900 filenames. No more going to DOS to see whats on each disk. Run files. LOAD files. A must for all programmers.

Price £18.95

BASIC COMMANDER

Single keystrokes allow you to LIST. SAVE, ENTER, LOAD or RUN files. You never need to type DOS again. View the entire disk directory. RENAME, LOCK, UNLOCK, even FORMAT disks from BASIC. Automatic line numbering, block delete, re-numbering. Also 3 keys for you to program.

Price £24.95

GAMES

ADVENTURE INTERNATIONAL		
The HULK	cass	7.95
The HULK	disk	17.95
BRODERBUND		
DROL	disk	24.95
LODE RUNNER	disk	24.95
SPARE CHANGE	disk	24.95
CALISTO		
WARLOCK	cass	14.95
WARLOCK	disk	14.95
CHANNEL 8		
GOLDEN BATON	cass	8.95
COSMI		
SLINKY	cass	8.95
SLINKY	disk	12.95
AZTEC CHALLENGE	cass	8.95
DATAMOST		
MR. ROBOT	disk	21.95
DATASOFT		
BRUCE LEE	cass	14.95
BRUCE LEE	disk	14.95
DALLAS QUEST	disk	14.95
FIRST STAR		
BOULDER DASH	cass	20.95
BOULDER DASH	disk	20.95
BRISTLES	cass	20.95
BRISTLES	disk	20.95
FLIP FLOP	cass	20.95
FLIP FLOP	disk	20.95
FUNSOFT		
FLAK	disk	14.95
SNOKIE	cass	10.95
SNOKIE	disk	13.95

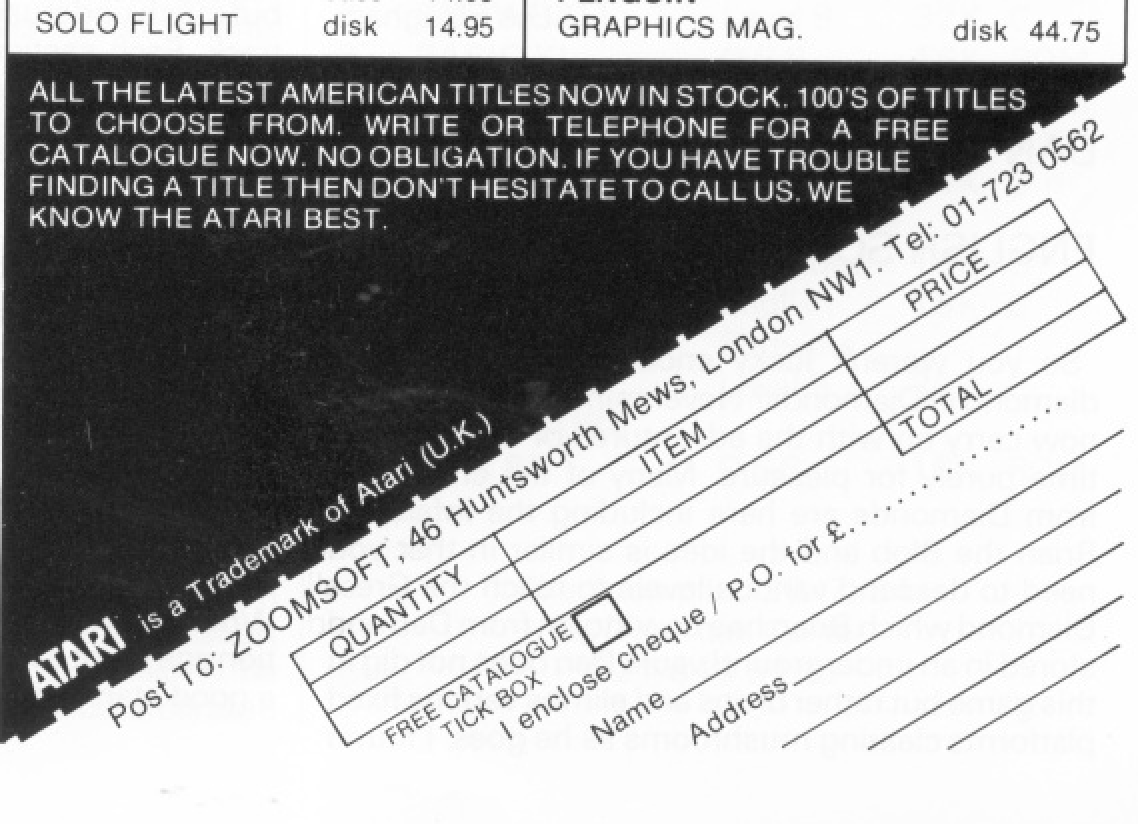
GAMES

GAMESTAR		
STAR LEAGUE B/B	disk	20.95
INFOCOM		
ZORK 1	disk	24.95
ZORK 2	disk	24.95
ZORK 3	disk	27.95
LIAMASOFT		
GRIDRUNNER	cass	7.50
MICROPROSE		
SPITFIRE ACE	cass	9.95
SPITFIRE ACE	disk	12.95
SOLO FLIGHT	cass	14.95
SOLO FLIGHT	disk	14.95

UTILITIES

O/SYSTEMS ACTION	cart	66.95
MAC/65(OS/A+)	disk	57.95
BASIC XL	cart	66.95
MMG		
BASIC DEBUGGER	disk	25.95
BASIC COMMANDER	disk	25.95
MAIL LIST	disk	34.95
TRONIX		
S.A.M.	disk	41.95
P.M. ANIMATOR	disk	25.95
TELETARI	disk	27.95
PENGUIN		
GRAPHICS MAG.	disk	44.75

ALL THE LATEST AMERICAN TITLES NOW IN STOCK. 100'S OF TITLES TO CHOOSE FROM. WRITE OR TELEPHONE FOR A FREE CATALOGUE NOW. NO OBLIGATION. IF YOU HAVE TROUBLE FINDING A TITLE THEN DON'T HESITATE TO CALL US. WE KNOW THE ATARI BEST.



BLUE THUNDER 16/48K CASS.

RICHARD WILCOX 1 PLAYER

Blue Thunder is a helicopter rescue mission similar to Choplifter but if you have not seen that game or could not afford it, you can get similar enjoyment from Blue Thunder for just £9.95. Your mission is to rescue your comrades from a remote island whilst avoiding hazards such as ground and sea based missiles, armoured barrage balloons and electric storms. Your jetcopter is armed and can destroy the nuclear reactor besides which your comrades are held captive but not the gun emplacements which you must pass. The best feature of the game is the control of the helicopter which is quite realistic (a fact confirmed by a reader in the RAF!) and is fairly complex to master. The joystick controls the direction of the helicopter in the normal left, right, up and down motion but with the trigger pressed, the helicopter can be turned. A medium press will turn through 90° and a long press through 180°. As the trigger is also used to fire missiles (via a short press) it can take some time to master the controls. The helicopter banks as it turns and heads away and looks very good. The background graphics are of reasonable standard and the game scrolls over six screens. In the short time available for review I was not able to rescue anybody so there would appear to be some challenge to the game which should keep you occupied for many hours. Blue Thunder is not a fast paced game but requires plenty of control and is a worthy addition to your collection.

Les Ellingham

DAN STRIKES BACK 16K

ENGLISH SOFTWARE 1 PLAYER

So you weren't lucky enough to win the real diamond in Diamonds? Never mind you can at least now carry on with the adventures of Dan but this time purely for pleasure. Many of the characters from Diamonds are here including the infamous Brian the Blob and the idea is similar in that you need to descend various levels to reach the Great Diamond which Brian has now stolen from Dan and stored in an underground vault. Dan does not dig in this game but rather drops and climbs various fixed platforms clearing mushrooms as he goes. I found Dan Strikes Back to be much more interesting than Diamonds as there is more thought required to work out the correct directions to clear each level. Various sections have gates which open and close making it necessary to work out moves in advance to ensure that you do not get trapped. One of the problems is that Brian is also in the vault and follows you everywhere.

Before you can gain the next level you must cover and clear each part of the current level and you have the chance to score bonus points by collecting magic toadstools which occasionally spring up. On level 3 a giant spider prowls the vault leaving webs which you must break through and which delay your progress. The puzzles of how to reach each part of the vault get more complex at each of the six levels until finally you can see the Great Diamond. All you have to do then is get back out again with Brian and all his nasty friends in hot pursuit. If you enjoyed Diamonds you should also enjoy Dan Strikes Back. I certainly found it more satisfying.

Les Ellingham

GEOMETRY 16K CASS.

SOFLOW SOFTWARE

Geometry is the first in a series of educational software from Soflow Software designed as a revision aid for the 'O' level syllabus. The program comes in several sections and uses not only diagrams and text on screen but also that unique, but much underused, Atari feature, the audio sound track. Each section of the program is introduced and commented on by a clear female voice on the soundtrack and care has been taken to ensure that the soundtrack supplements the program and does not merely repeat what is on screen. The program itself requires interaction from the user in answering questions at various stages. Unfortunately the subject itself was too complex for me(!) but I have been assured that the content has been checked for accuracy by a qualified teacher so there should be no complaint in this area.

Many parents have bemoaned the lack of educational software for the Atari and the little that has been published is mainly for pre-school children. Writing software for children of secondary education age is not easy but the Softswot series provides a good start in this neglected area.

Les Ellingham

MUSIC CONSTRUCTION SET

ELECTRONIC ARTS 48K DISK

In the wake of so many graphics programs and utilities, the sound and music capabilities of Atari computers seems to have been ignored, but now Music Construction Set from Electronic Arts has put that to rights.

The screen consists of two staves (the Bass and Treble clefs) and a graphic menu. The menu contains a complete set of notes and rests, a time signature indicator, a time counter and nine symbolic pictures, or icons, representative of a certain command, e.g. a disk for disk commands. The approach is very similar to Pinball Construction Kit. The Hand icon is the most important as this contols the entire workings of the MCS. The hand can be controlled from a choice of controllers such as joystick and keyboard or the Atari Touch Tablet or Koala Pad.

The hand is used to control the speed, sound and volume, time signature and for manipulation of the icons and even to change the key in which the music is played. Most importantly, however, it is used to write music. By placing the hand on a note, rest or other musical notation such as ties, octave raisers etc., and pressing the trigger, the item can be picked up and placed anywhere on the staves. The facility is ideal for music development or copying from manuscripts.

In the middle of the screen are five gauges which give complete control over speed, volume and various instrument effects such as piano, drum, accordian and vibrato. The editing facilities are also extremely efficient using the 'cut and paste' technique. Bars can be cut out and replaced anywhere in the same piece of music or even another piece loaded from disk. It is also possible to print the music on a dot-matrix printer giving a full printout of the score.

There are drawbacks, however, to what is otherwise an excellent program. In the disk command mode, if load or save are not specified and the filename begins with F, the program will format any disk in drive 1 without chance of verification. Secondly, it can only take 70 bars of music from either stave in memory at once which is a little short and might not allow a full piece of music to be worked on at any one time. Thirdly the controls are a little coarse and take practice to operate correctly.

The MCS is set up to use only three of the Atari's four voices in order to give better bass notes but you can select four voices if required. It is not only the bass notes that sound good however, the whole musical quality must be heard to be believed.

Gary Sabin and Julian Bailey

DUELLIN DROID 16K

ENGLISH SOFTWARE 1 PLAYER

If you are looking for a fast action robot-shooting game, give Duellin' Droid a look for it certainly packs some fast action. The game itself is very simple, all you need to do is shoot all of the creatures in the arena and score bonus points by rescuing the "Little Pink People" who wander about. The main appeal is the speed and there are 99 levels to survive. Graphically it is very strange, written possibly in Antic mode 5(?), and it looks a little disappointing at first. Once into the game however this is not so important, it is just a question of surviving those 99 levels.

Les Ellingham

TOP TEN

1	**ENCOUNTER**	Novagen	16K	C
2	**SNOWBALL**	Level 9	32K	C
3	**BASIC XL**	O.S.S.	16K	ROM
4	**ACTION!**	O.S.S.	16K	ROM
5	**BRUCE LEE**	Datasoft	32K	C/D
6	**THE HULK**	Adventure International	24K	C
7	**DANGER RANGER**	Microdeal	16K	C
8	**A.C.E.**	English Software	16K	C
9	**COLOSSAL ADVENTURE**	Level 9	32K	C
10	**MAC 65**	O.S.S.	16K	ROM

Chart compiled 20/7/84
by The Atari Center
021 643 9100

Utility

VARSORT 1

This is the first of two Variable utility programs. The second VARSORT2 which allows you to add descriptions to the variables and which can be accessed from this program will be published next issue. Whilst written for disk users, the program can be easily converted for cassette.

When writing a program which uses quite a lot of variables, it is very easy to forget which variables you have already used. It is possible to obtain a list of these from the computer itself but they will be in the order of entering. This program is 'ENTERed' from disc and tacks itself on to the end of your program (its line numbers run from 32000 upwards). When 'ENTERed' type 'GOTO 32000' and the program will clear the screen and print, in three columns, all the variables you have used so far. You will then be asked to press any key for a 'sort'. The sort section is the one used in a program published in Page 6 some time ago although in this version you can watch its progress on the screen. When completed, the sorted list of variables is displayed on screen, again in three columns. You are then asked if you want to print them or put them to disk - the filename is fixed. Finally you have the option to call 'VARSORT2' or finish. A word of warning - whichever option you choose will erase your original program from memory - so SAVE IT FIRST !!!.

The program runs as follows:-

FETCHING THE VARIABLES

Line 32000 dimensions the strings and array used. The array is cleared.
Line 32010 sets the mode to zero: sets the colours: clears the cursor.
Line 32020 prints the heading asking you to wait for the variables to be fetched.
Line 32030 sets a TRAP: OPENs a channel for input from keyboard.
Line 32040 sets a loop 'XX' which will run from the address contained within memory locations 130 and 131 to that in 132 and 133 *LESS 47*. The first is the beginning of the variable list - the last is the end minus 1. Why 47 then? The variables used in this program occupy 46 bytes and, as we don't want them listed we stop the list just before they are reached.
(A word of explanation - the variables are listed nose to tail with no gaps in between. What makes it easy to pick out each seperate variable is the way in which the last character is stored - as an INVERSE character. The ATASCII number for an Inverse Character is 128 greater than that for a normal one).
Line 32050 looks at the value contained in 'XX', if less than 128 then its value is passed on to 'XXX': GOSUB's line 32100 and on returning GOTO's 32070.
Line 32060 is only reached when a PEEK at 'XX' is greater than 128 and when this is so then 'XXX' is made equal to 'XX' but less 128: GOSUB's 32100 and, on return, is immediately GOSUBbed again to line 32110, for it obvious that a complete variable name has been obtained.
Line 32070 sends you back for another pass through the loop until finally the list is exhausted whereupon the program GOSUB's to line 32440 (the screen print-out).
Line 32080 prints the request to press any key for the sort to begin and then waits for your input.
Line 32090 clears the screen and GOTO's line 32150.
Line 32100 adds the individual characters of the variable (in 'XXX') to a string 'XX$'.
Line 32110 is brought into use when the complete variable is contained in 'XX$', the string is then checked for length, if not 10 characters then it has a dot added until it is. As the final string will contain all the variables, each sub string must be of a fixed length so that we can extract it!
Line 32120 adds 'XX$' to that final 'long' string, 'XAR$' and 'XX$' are then cleared ready for the next variable.
Line 32130 is the error line which the TRAP, when sprung, sends the program to.
Line 32140 sends you back to line 32010 to have another try.

THE SORT

Line 32150 is the first line of the sort section. The bell is rung, warning you of the start: 'XX' is used again, this time to represent the length of each variable sub string. A loop is started - 'XZX1' - running from 1 to the length of 'XAR$' divided by 'XX', this of course gives you the number of variables in the string: 'XX1' is equalled to the pass through the loop 'XZX1' times 'XX': 'XROW' is equalled to the number of variables in the string.
Line 32160 Another loop 'XZX2' is started running from 1 to the number of variables: 'XX2' is equalled to 'XZX2' times 'XX'. This second loop completes a whole cycle whilst the first loop is stationary on one item at a time. This is used by the line 32180.
Line 32170 prints the loop positions as they are executed.

continued on page 38

a variable sorting utility by C.L.Stone

```
31995 REM ***************************
31996 REM *       VARSORT 1         *
31997 REM * BY C.L. STONE 14/1/84 *
31998 REM ***************************
31999 REM
32000 DIM XX$(10),XAR$(1280).XAR1$(128
0),XROW(128):XROW(1)=0:XROW(128)=XROW:
XROW(2)=XROW
32010 GRAPHICS 0:POKE 710,16:POKE 712.
130:POKE 709,26:POKE 752,1
32020 POSITION 9,1:? "VARIABLES BEING
FETCHED                    PLEASE WAIT
"
32030 TRAP 0:TRAP 32130:CLOSE #1:OPEN
#1,4,0,"K:"
32040 FOR XX=PEEK(130)+PEEK(131)*256 T
O PEEK(132)+PEEK(133)*256-47
32050 IF PEEK(XX)<128 THEN XXX=PEEK(XX
):GOSUB 32100:GOTO 32070
32060 XXX=PEEK(XX)-128:GOSUB 32100:GOS
UB 32110
32070 NEXT XX:PRINT :GOSUB 32440
32080 POSITION 5,22:? "PLEASE PRESS AN
Y KEY FOR SORT      ":GET #1,XXX
32090 ? CHR$(125):GOTO 32150
32100 XX$(LEN(XX$)+1)=CHR$(XXX):RETURN

32110 IF LEN(XX$)<10 THEN XX$(LEN(XX$)
+1)=".":GOTO 32110
32120 XAR$(LEN(XAR$)+1)=XX$:XX$="":RET
URN
32125 REM > FIRST THREE CHARACTERS BEL
OW ARE ESCAPE,CONTROL "2"
32130 ? "   TRY AGAIN PLEASE - TYPE
 'CONT'   ":? "              AND PRESS <E
NTER>":STOP
32140 GOTO 32010
32150 ? CHR$(253):XX=10:FOR XZX1=1 TO
LEN(XAR$)/XX:XX1=XZX1*XX:XROW=LEN(XAR$
)/XX
32160 FOR XZX2=1 TO LEN(XAR$)/XX:XX2=X
ZX2*XX
32170 POSITION 2,10:? "1st. SORT ";XZX
1;"  ";XZX2;"  "
32180 IF XAR$(XX1-9,XX1)<XAR$(XX2-9,XX
2) THEN XROW=XROW-1
32190 NEXT XZX2
32200 XROW(XZX1)=XROW
32210 NEXT XZX1
32220 FOR XZX1=1 TO LEN(XAR$)/XX:XX1=X
ZX1*XX
32230 FOR XZX2=1 TO LEN(XAR$)/XX:XX2=X
ZX2*XX
32240 POSITION 21,10:? "2nd. SORT ";XZ
X1;"  ";XZX2;"  "
32250 IF XZX1=XROW(XZX2) THEN XAR1$(XX
1-9.XX1)=XAR$(XX2-9.XX2)
32260 NEXT XZX2
32270 NEXT XZX1
32280 XAR$="":XAR$=XAR1$:XAR1$=""
32290 POSITION 13,12:? "SORT COMPLETE
D ":REM 1ST.CHARACTER ESCAPE. CONTROL"
2", WORDS INVERSE
32300 FOR XXX=1 TO 500:NEXT XXX:? CHR$
(125):GOSUB 32440
32320 POSITION 3,22:? "DO YOU WISH TO
PRINT THE LIST Y/N ":GET #1.XXX:IF XXX
=78 OR XXX=110 THEN 32380:REM Y/N INV.
32330 TRAP 0:TRAP 32340:LPRINT " ":GOT
O 32350
32340 POSITION 5,22:? "   SWITCH ON T
HE PRINTER PLEASE ":REM 1ST. CHARACTER
 ESCAPE,CONTROL"2"
32345 GOTO 32330
32350 LPRINT CHR$(15):CHR$(27):CHR$(48
)
32360 XX=10:FOR XZX1=1 TO LEN(XAR$)/XX
:XX1=XZX1*XX:LPRINT XAR$(XX1-9.XX1):NE
XT XZX1:LPRINT CHR$(27);"@"
32370 REM > Y/N IN LINES 32390 AND 324
30 ARE IN INVERSE VIDEO
32380 POSITION 0,22:? " DO YOU WISH T
O RECORD ON DISC  Y/N   ";:GET #1,D:IF
 D=78 OR D=110 THEN 32420
32390 TRAP 32410:CLOSE #3:OPEN #3,8,0.
"D:XXX.DAT"
32400 PRINT #3:XAR$
32410 CLOSE #3
32420 POSITION 0,22:? " DO YOU WISH TO
 CALL VARSORT2  Y/N ":GET #1.D:IF D=8
9 OR D=121 THEN CLOSE #1:RUN "D:VARSOR
T2.SAV"
32430 CLOSE #1:NEW
32440 XX=10:XXXLIN=5:XZX1=0:XZX2=1
32450 FOR XX2=1 TO LEN(XAR$)/XX:XX1=XX
2*XX:POSITION XZX1.XXXLIN:? XAR$(XX1-9
.XX1):XXXLIN=XXXLIN+1
32460 IF XXXLIN=19 AND XZX2=1 THEN XXX
LIN=5:XZX1=14:XZX2=2:GOTO 32490
32470 IF XXXLIN=19 AND XZX2=2 THEN XXX
LIN=5:XZX1=28:XZX2=3:GOTO 32490
32480 IF XXXLIN=19 AND XZX2=3 THEN XXX
LIN=5:XZX1=0:XZX2=1:GOSUB 32500
32490 NEXT XX2:RETURN
32500 POSITION 3.22:? "PLEASE PRESS AN
Y KEY FOR NEXT PAGE":GET #1.XXX
32510 FOR XXX=19 TO 5 STEP -1:POSITION
 0,XXX:? "
                 ";:NEXT XXX:RETURN
```

□

VARSORT continued from page 36

Line 32180 compares the variable in the first loop (using 'XX1' to split it from the long string) with the variables as they appear in the second. When the first variable is 'less' (alphabetically speaking!) than that in the second loop then 'XROW' (initially representing the total number of variables) is reduced by one.
Line 32190 goes back for another pass through loop 'XZX2' until this is completed.
Line 32200 comes into effect when loop 'XZX2' has completed its cycle, checking each of its items against the single item of loop 'XZX1'. 'XROW' now holds the actual position which the loop 'XZX1' item will hold in the new string. This is now placed in the array 'XROW', the position within the array being determined by the pass through loop 'XZX1'.
Line 32210 now returns for the next pass through loop 'XZX1' (just one item before going through loop 'XZX2' again). 'XROW' is automatically re-initialized to the total number of variables before the loop 'XZX2' starts. This continues until the whole of loop 'XZX1' has been compared with the items shown by loop 'XZX2', and the 'XROW' array contains the new positions.
Line 32220 starts a loop 'XZX1' running as before. 'XX1' is set as before.
Line 32230 starts loop 'XZX2' as before. 'XX2' is set as before.
Line 32240 prints the state of the loops to screen (alongside the other).
Line 32250 checks the first loop 'XZX1' and if the pass is equal to 'XROW' array as defined by the pass through the loop 'XZX2' then the new string has the variable shown in that pass through 'XZX2' transferred to the new string in the position which 'XZX1' and 'XROW (XZX1)' agree upon. Line 32260 goes back for another loop through 'XZX2' and line 32270 through 'XZX1'.
Line 32280 When all is finished and the new string 'XAR1$' contains the sorted list then this line sets about restoring the original string. 'XAR$' is first emptied, then 'XAR$' is made equal to 'XAR1$', finally 'XAR1$' is emptied.
Line 32290 prints the fact that the sort has been completed and line 32300 pauses for a while, clears the screen and GOSUB's line 32440 for a print out.

PRINT IT?

Line 32320 asks you if you wish to print the list - waits for an answer Y or N. An 'N' would send the program to line 32380.
Line 32330 sets a TRAP in case the printer is not switched on: does an exploratory LPRINT, which if O.K. then passes on to line 32350.
Line 32340 is the error routine which prints the message to switch the printer on. Line 32345 sends you back to line 32330 so creating a loop, this will continue (ringing the bell in the process) until the printer is switched on.
Line 32350 sets the printer to the Condensed Print and 8 lines to the inch mode. This should be changed to suit your own printer.
Line 32360 sets 'XX' to equal 10: starts a loop 'XZX1' running from 1 to the number of variables as before: sets 'XX1' as before: prints to paper the variables out of XAR$ as defined by 'XX': goes back for another pass: clears the printer settings.
Line 32380 asks if you wish to record to Disc. An 'N' would send you to line 32420. The variables would be filed under the name D:XXX.DAT and can be used by the program VARSORT2.
Line 32390 sets a TRAP to close the channel when finished: OPENs a channel to write data to the above file.
Line 32400 prints the data to disc.
Line 32410 closes the channel.
Line 32420 asks you if you wish to call the program VARSORT2. A 'Y' would cause VARSORT2 to be run, so erasing both your own program and VARSORT1 in the process. The moral being SAVE IT!!
Line 32430 An 'N' would close the keyboard input channel and issue a NEW instruction, with the same effect as above with regard to your program.

PUT IT ON SCREEN

Line 32440 sets 'XX' to equal 10: 'XXXLIN' (line counter) to 5: 'XZX1' (screen print position) to zero.
Line 32450 starts a loop 'XX2' as before: 'XX1' is set as before: at a position defined by 'XZX1' and 'XXXLIN' the variables are printed: 'XXXLIN' has 1 added to it.
Line 32460 'XXXLIN' is checked, if 19 and 'XZX2' equals 1 the 'XXXLIN' is reset to 5: 'XZX1' to 14: 'XZX2' to 2: the program then GOTO's line 32490.
Line 32470 checks 'XXXLIN', if 19 and and 'XZX2' equals 2 then 'XXXLIN' is reset to 5: 'XZX1' to 28: 'XZX2' to 3: and goto's line 32490.
Line 32480 checks 'XXXLIN', if 19 and 'XZX2' equals 3 then 'XXXLIN' is reset to 5: 'XZX1' to zero: 'XZX2' to 1: and GOTO's line 32500. ▶

LANDSCAPE continued from page 22

```
1950 DATA 9,94,236,72,80,96,64,128
1960 DATA 1,2,4,8,16,32,64,128
1970 DATA 144,122,55,18,10;6,2,1
1980 DATA 128,64,32,16,8,4,2,1
1990 DATA 0,24,126,255,126,36,36,0
2000 DATA 0,0,24,24,24,24,24,0
2010 DATA 0,0,0,0,204,204,51,51
2020 DATA 42,148,75,186,93,146,41,84
2030 DATA 16,40,68,146,68,40,16,0
2040 DATA 0,130,146,170,214,170,146,13
0
2050 DATA 16,40,40,84,84,170,84,0
2060 DATA 255,255,129,66,36,24,92,231
```

Line 32490 sends you back for another pass through the loop. When the loop is finished the program RETURNS to a position immediately after the original GOSUB.
Line 32500 asks you to press a key for the next page and then waits for an input.
Line 32510 uses a loop 'XXX' to print a line of blanks from lines 19 to 5: returns to line 32490.

Even if you do not type it in I hope that the line by line explanation will help you to understand the structure of the program and will help you to write your own utilities. VARSORT2 will follow in the next issue.

FLAGS continued from page 30

```
1900 POKE A,70:POKE A+C3,C6:POKE A+4,C
6:POKE A+C5,C6
1910 POKE CR1,C12:POKE CR2,118:POKE CR
3,50:POKE CR5,24:GOSUB SCAN:RETURN
1920 GRAPHICS C18
1930 POSITION C2,C3:? #C6;"national fl
ags"
1940 POSITION C1,7:? #C6;"AUTO-DISPLAY
 IS"
1950 POSITION C1,9:? #C6;"NOW OPERATIN
G..."
1960 FOR F=C1 TO 20
1970 POKE CR1,118:FOR W=C1 TO C10:NEXT
 W
1980 POKE CR1,C0:FOR W=C1 TO C10:NEXT
 W
1990 NEXT F:RETURN
2000 GRAPHICS C0:POKE 82,C1:POKE CR2,C
0:POKE CR3,C10
2010 ? "     NATIONAL FLAGS ... Keith
Berry"
2020 ? " ┌──────┐ ┌─┬─┬─┐ ┌─┐
──────┘"
2021 ? " |      | | | | | | |
      |"
2022 ? " ├──────┤ | | | | | |
      |"
2023 ? " |      | | | | | | ├
──────┤"
2024 ? " ├──────┤ | | | | | |
      |"
2025 ? " |      | | | | | | |
      |"
2026 ? " └──────┘ └─┴─┴─┘ └─┘
──────┘"
2100 ? "Very many National Flags consi
st of a  pattern of just three colours
 in one"
2110 ? "of the above designs, with or
without":? "the addition of an emblem.
   This"
2120 ? "program was devised to aid in
their    identification."
2130 ? :? "Scandinavian Cross flags ar
e included, together with a few others
 to add some variety.":?
2140 ? "You will have the option of a
Quiz or  a continuous display."
2150 ? :? " Press any key to continue.
..       ";
2160 Z=PEEK(INK):IF Z=C255 THEN 2160
2170 POKE INK,C255:RETURN
```

Hardware

THE HARD(WARE) FACTS

EVERYTHING YOU WANTED TO KNOW ABOUT YOUR ATARI BUT WERE AFRAID TO ASK (PART 3)

HEADPHONES

Ever thought about monitoring the sound produced by your computer on headphones? This is okay if you have a TV with a headphone socket but it is not really hi-fi because it is usually wired by the manufacturers into the speaker circuit and any hiss or hum present is more noticeable. So what can you do to improve quality or if you do not have a TV with a headphone socket? Firstly you can forget any ideas you might have of adding a socket to a TV which was not designed to have one. Definitely not on. You have a choice of simply connecting a small amplifier to the audio output of the computer externally and letting this drive your headphones as shown in Figure 4 or fit a small special purpose amplifier inside the computer with a socket for headphones fitted in a convenient position. The latter is of course the neatest solution but requires more work and involves drilling holes and soldering. The choice is yours.

To prevent any problems with possible leaky batteries it is suggested that batteries are not put inside the computer. Possibly a plug and socket arrangement at the rear or a seperate low voltage power supply unit driven from the mains. Such a project is for experienced constructors only though.

SERIAL I/O PORT

One of the lesser known parts of the Atari is the peripheral connector at the side into which your 410 or other peripherals plug. It is a 13 pin Molex plug, part number MX 7514-4.13 - a non-standard plug if ever I saw one! A pity Atari did not put on a nice standard D-type connector.

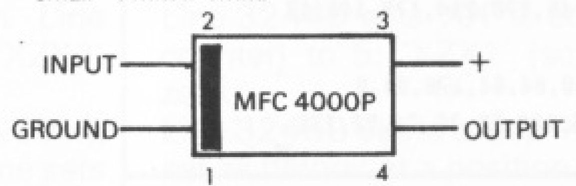
Figure 1.

2 4 6 8 10 12
1 3 5 7 9 11 13

1. CLOCK I/P	5. DATA O/P	9. PROCEED
2. CLOCK O/P	6. GROUND	10. +5V READY
3. DATA I/P	7. COMMAND	11. AUDIO I/P
4. GROUND	8. MOTOR CTL	12. +12V
	13. INTERRUPT	

Not only is the number of pins non-standard, the diameter and spacing is also different so any ideas you may have about cutting down a 15-way D-type socket simply will not work, I have tried it. If you want to plug anything in to this socket, you have a problem. I have not yet been able to find a supplier in the U.K. for the socket to fit the 13-way I/O plugs but you can buy these sockets with any length of shielded (screened) cable attached from Elite Digital, P.O. Box 1414, Melville, New York 11747. For those of you who are interested in trying expansions through the I/O port details of the connections are given in Figure 1.

The MFC4000P I/C chip used for the amplifier gives 250mw(¼ watt) with 9 volt(max.) supply. More than enough to drive headphones. It is very small - 7mm x 4mm.



Unfortunately you need almost a dozen extra parts to make it work, but then that applies to most chips. The complete circuit diagram is shown in Figure 5.

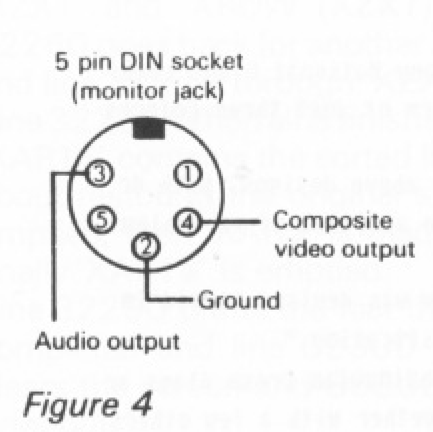


Figure 4

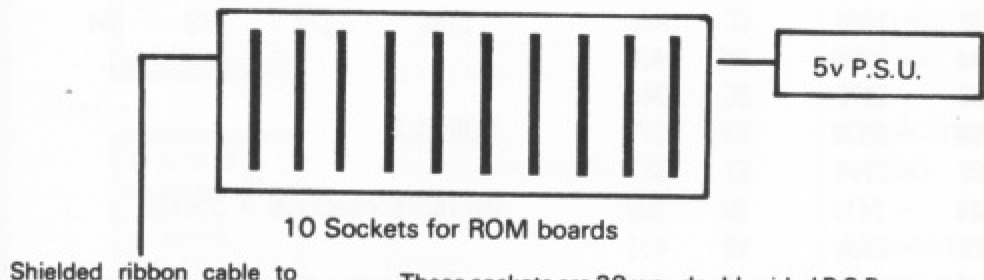
Figure 5

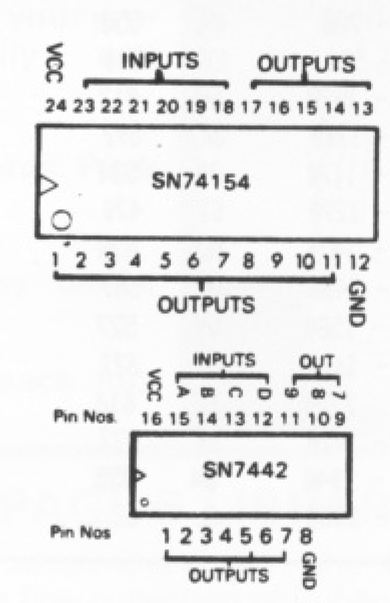
GND to Pin 2 · To Pin 3 · Volume 10K · 4.7µf · 1K · MFC 4000P · 250µf · ON/OFF SWITCH · 4.7K · 0.003µf · 10K · 5000PF · Loudspeaker or headphones socket · 100µf · 9 volt battery

A ROM EXPANSION BOX

The advantage of games cartridges is that they instant load, however you still have to open the lid to change cartridges. With tape or disk it is possible to load a particular game by typing in its name and the computer will search the tape or disk for the program required. It would be nice to select cartridges in this way but of course you can only have one cartridge inserted at a time. Even on the 800 the second slot is of no use in this connection.

The solution is, I think, to have a seperate box containing its own low voltage power unit and say 10 sockets wired in parallel, except for the CS pins which are individually wired. A multiway ribbon cable is then fitted with a suitable connector and plugs into the left hand ROM socket of the computer. It should be possible for someone to write a short program and with the aid of an EPROM burner put it on ROM which would be put in slot 1 of the expansion box. This ROM would contain a menu. The BASIC cartridge would be put into slot 2 with various other games or languages taking up the remaining slots. The Menu program would ask which of the available cartridges you wished to load and exiting a program by using SYSTEM RESET would bring you back to the Menu for a new selection. I thought of this idea about six months ago and keep meaning to try it out but available time does not permit. On this occasion perhaps I can pass the idea over to PAGE 6 readers and maybe one of you could come up with a finished product or prototype. Figures 2 and 3 give a little more information to get you started.

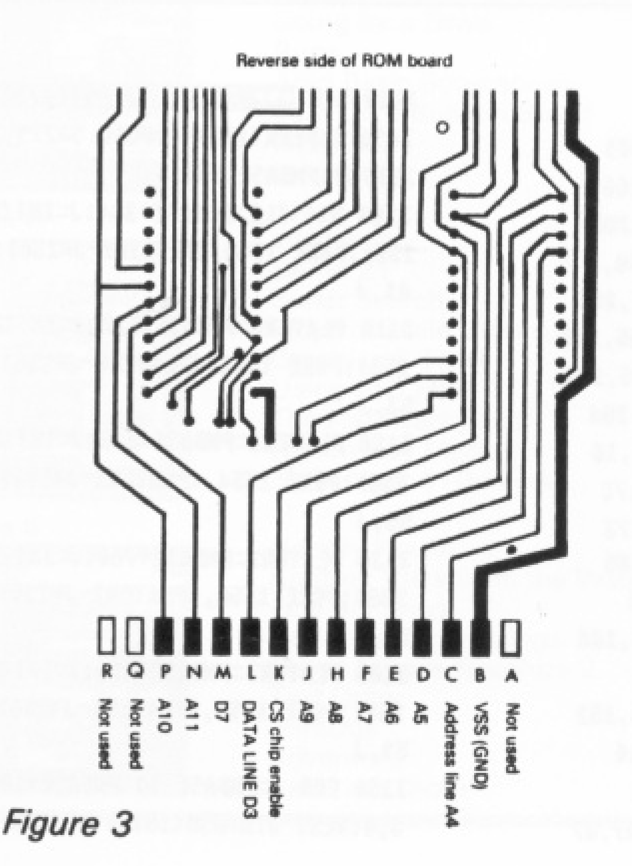




These sockets are 30 way double sided P.C.B. connectors part no. AMP-8006-1-530831-5. They are wired in parallel with the exception of the CE chip enable pin. It is suggested that a standard TTL I/C such as 7442 is used to bring the appropriate pin low.

If you want to install 16 sockets instead of the suggested 10 then I would suggest using 74154 TTL I/C.

Figure 2



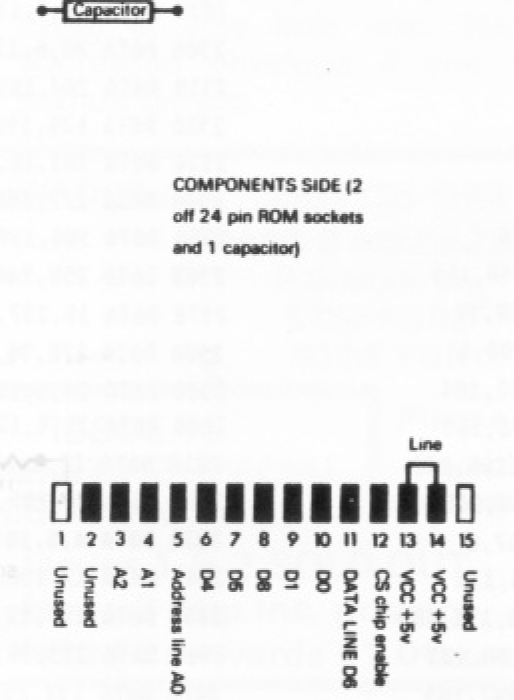


Figure 3

TYPO TABLES

FULL DETAILS OF TYPO APPEAR IN ISSUE 5

COLOURFLOW

Variable checksum = 166718

Line num range	Code	Length
1000 - 1110	IY	332
1120 - 1230	KB	449
1240 - 1330	AQ	299

LANDSCAPE

Variable checksum = 981791

Line num range	Code	Length
0 - 100	DT	517
110 - 190	TA	531
200 - 290	JQ	507
300 - 410	HS	392
420 - 500	LG	509
510 - 580	UJ	547
590 - 700	GM	472
710 - 780	NA	550
790 - 900	EH	510
910 - 1020	GQ	412
1030 - 1100	JF	592
1110 - 1170	JS	534
1180 - 1290	SI	470
1300 - 1410	MJ	430
1420 - 1480	HK	587
1490 - 1580	VY	527
1590 - 1670	QL	622
1680 - 1790	CU	616
1800 - 1920	JX	477
1930 - 2040	GN	355
2050 - 2060	JL	62

GRID

Variable checksum = 988869

Line num range	Code	Length
10 - 140	MG	419
150 - 260	QT	429
270 - 370	HA	562
380 - 480	CV	538
490 - 580	HU	526
590 - 700	HA	503
710 - 810	DD	583
820 - 910	TE	554
920 - 980	RD	525
990 - 1090	QL	555
1100 - 1210	RS	444
1220 - 1300	MZ	502
1310 - 1420	MG	424
1430 - 1520	IF	535
1530 - 1600	ZL	503
1610 - 1710	YO	557
1720 - 1830	CC	498
1840 - 1950	GW	430
1960 - 2070	ZC	308
2080 - 2170	CJ	579
2180 - 2290	SI	397
2300 - 2410	HX	337
2420 - 2530	YR	411
2540 - 2650	JT	310
2660 - 2770	TK	193
2780 - 2890	CL	255
2900 - 3010	CO	274
3020 - 3130	MW	514
3140 - 3150	VT	123

FLAGS

Variable checksum = 3530505

Line num range	Code	Length
0 - 35	CG	432
36 - 125	XE	466
130 - 300	IK	354
310 - 401	CG	358
402 - 413	RK	450
414 - 425	HE	466
426 - 437	DW	449
438 - 850	SE	425
860 - 940	PJ	440
950 - 1060	CU	305
1070 - 1180	MS	180
1190 - 1300	OJ	271
1310 - 1420	OP	294
1430 - 1540	DA	299
1550 - 1660	VE	338
1670 - 1780	SS	265
1790 - 1900	WB	472
1910 - 2020	VJ	362
2021 - 2130	FP	599
2140 - 2170	KB	154

TURTLE

Variable checksum = 586956

Line num range	Code	Length
1 - 100	TU	520
101 - 1050	RD	579
1060 - 1130	GW	556
1140 - 2000	GH	610
2030 - 2070	MC	340

GRID continued from page 12

```
2700 DATA 0,0,0,0,0
2710 DATA 0,0,0,0,0
2720 DATA 0,0,0,0,0
2730 DATA 0,0,0,0,0
2740 DATA 0,0,0,0,0
2750 DATA 0,0,0,0,0
2760 DATA 0,0,0,0,104
2770 DATA 162,6,160,90,169
2780 DATA 7,32,92,228,96
2790 DATA 173,26,6,208,6
2800 DATA 32,187,6,32,104
2810 DATA 6,76,98,228,169
2820 DATA 3,133,203,166,203
2830 DATA 189,0,6,133,205
2840 DATA 188,8,6,157,8
2850 DATA 6,189,75,6,133
2860 DATA 204,138,10,170,189
2870 DATA 16,6,133,206,189
2880 DATA 17,6,133,207,189
2890 DATA 27,6,133,208,189
2900 DATA 28,6,133,209,166
2910 DATA 204,169,0,145,206
2920 DATA 136,192,255,240,3
2930 DATA 202,16,246,164,204
2940 DATA 177,208,164,205,145
2950 DATA 206,136,132,205,192
2960 DATA 255,240,4,198,204
2970 DATA 16,237,198,203,16
2980 DATA 178,96,162,3,173
2990 DATA 24,6,133,206,173
3000 DATA 25,6,133,207,188
3010 DATA 12,6,189,231,6
3020 DATA 49,206,145,206,188
3030 DATA 4,6,189,227,6
3040 DATA 17,206,145,206,152
3050 DATA 157,12,6,202,16
3060 DATA 229,96,2,12,32
3070 DATA 192,252,242,207,47
3080 REM *** P/M INITIALIZATION ***
3090 I=PEEK(106)-8:POKE 54279,I:POKE 53277,3:PMBASE=I*256
3100 MISSILE=PMBASE+384:J=INT(MISSILE/256):POKE 1560,(MISSILE-J*256):POKE 1561,J
3110 PLAYER0=PMBASE+512:J=INT(PLAYER0/256):POKE 1552,(PLAYER0-J*256):POKE 1553,J
3120 PLAYER1=PMBASE+640:J=INT(PLAYER1/256):POKE 1554,(PLAYER1-J*256):POKE 1555,J
3130 PLAYER2=PMBASE+768:J=INT(PLAYER2/256):POKE 1556,(PLAYER2-J*256):POKE 1557,J
3140 PLAYER3=PMBASE+896:J=INT(PLAYER3/256):POKE 1558,(PLAYER3-J*256):POKE 1559,J
3150 FOR J=PMBASE TO PMBASE+1024:POKE J,0:NEXT J:A=USR(1615):RETURN
```

■

BACK ISSUES

Issue 3



Calendar
Cricket Maths
Arcade Action
Character Redefinition
Character Generation Utility
Keyboard Techniques
Character Designer
Software Reviews
Master Directory

Issue 4



Lunar V
Arcade Action
Merlin's Magic Square
Memory Mapped Screens
Basic Timing
Grab an Apple
Software Reviews
Disk Sort
First Steps

Issue 5



Target
Memory Mapped Screens
Squares
Arcade Action - Miner 2049er
Vertical P/M Movement
Software Reviews
First Steps
Colour Selector
Line Lister

Issue 6



Memories
TeleCommunicate
Scramble
Time for Music
Dodger
Book Reviews
Hypnosis
Automatic Drive
First Steps

Issue 7



Slots
Seasons Greetings
Grubs
Going for a Drive
Bugs
Atari Basic Sourcebook
Your Own Bulletin Board
First Steps
Largeprint

Issue 8



Wildwest
Demo 21
Sonar Search
Player Missile Graphics
Graphics 8 Text
Using XIO FILL
The Hardware Facts
Return Key Mode
Make Your 410 Work!

Issue 9



Hungry Horris
Another Brick from the Wall
Fine Scrolling
Understanding Strings
Player Missile Graphics 2
Text Draw
MiniDos
Adventure
...and lots more!

DUST COVERS

Don't risk a breakdown of your precious equipment through dust and dirt! Protect all of your equipment when not in use with a custom made dust cover. Natural PVC with a brown trim - easily folded away when your equipment is not in use. Easily fitted with all peripherals, joysticks etc. in place.

400/800/600XL/800XL	£2.95
410 Old Style cassette*	£1.95
410 New Style cassette	£1.95
1010 cassette	£1.95
810/1050 disk drive	£2.95
1020/1027 printers	£3.95

*old style does not have a pause button.

Also for all popular printers (normally £3.95)

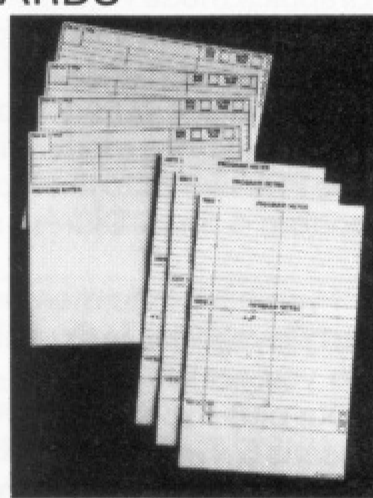
DISK/CASSETTE CARDS

Keep track of your programs easily and tidily.

Disk insert cards. Pack of 25

Cassette inlay cards. Pack of 25.

£1.00 each pack inc. postage.



THE PAGE 6 LIBRARY

Available to subscribers only, the **PAGE 6 LIBRARY** is a fine collection of Public Domain programs from around the world. Hundreds of user-written programs (including machine-code) at very little cost. Send a stamped addressed envelope if you have not had current details.

ORDER ALL ITEMS ON THIS PAGE FROM
PAGE 6 MAGAZINE
P.O.BOX 54,
STAFFORD
ST16 1DR

Please make cheques payable to PAGE 6.

ISSUE 10 - ADVENTURE SPECIAL - ALSO AVAILABLE

Issues 1 & 2 SOLD OUT
Issue 3 70p inc. postage
Issue 4 onwards £1.00 each inc.postage

Beginners

FIRST STEPS

Mark Hutchinson

Have you ever written a program in Graphics 1 with a text window and wished that you could have the space of Graphics 0 but still use the window? Well, you can. Normally changing the screen means altering the display list but as the window itself is in Graphics 0 I would consider this an impossibility. You may be able to use the vertical blank interrupt but you would need to know machine language. The answer is really very simple. When you call a Graphics mode the machine must store either 24 lines of Graphics 0, 4 lines for the window of another mode or 0 lines for a Graphics mode+16 (i.e. no text window). Location 703 is used for this purpose. All you need to do is call GR.0 and POKE 703,4. When the display handler reads 703 it is forced to open device #6 as with any other mode with a window. This means you PRINT #6; to the top 20 lines and PRINT to the bottom 4 lines. Didn't I say it was simple?

If you have little memory to spare and want to try out Player Missile Graphics in single-line resolution do you realise that you can only use five players (four plus the combined missiles) maximum which means 5 x 256 bytes? As you need to start on a 2K boundary, you may think that there are 3 x 256 bytes wasted. Not at all. You can still store other bits of data here if you wish but, be warned, BASIC A+ uses this technique so be careful if you use this language.

Have you ever wished to add a utility to your program but cannot as both the utility and the program use the page 6 storage area? The answer is simple. Set one or both routines up as a string. For example

```
10 FOR I=0 TO 30: READ DATA: POKE
1536+I,DATA: NEXT I
20 X=USR(1536)
```

uses page 6. Try the following

```
10 DIM A$(31): READ DATA: A$(I,I)=CHR$-
(DATA): NEXT I
20 X=USR(ADR(A$))
```

Although you have used extra memory through the DIM statement, the routine itself has now become relocatable, i.e. the computer protects the string itself and will move it around in memory to do so. Alternatively you could use

```
10 DIM A$(31): FOR I=0 TO 30: READ DATA:
POKE ADR(A$)+I,DATA: NEXT I
```

The FOR/NEXT - READ statements will take up a lot of space, so I suggest that once you have the string defined, delete the above lines and use the following after having RUN line 10 only.

PRINT A$ *(in direct mode)*
Use the editing features to make 19 spaces before the printed string and add
10 DIM A$(31): A$"
After the printed A$ add - ":X=USR(ADR(A$))

You can now delete all the DATA and save yourself some memory.

By taking this one stage further you could use X=USR(ADR(" *assembly codes* ")) and save eight bytes by not using a string. By coincidence you can also save eight bytes each time you use POKE instead of SETCOLOR and, finally, using cursor control characters takes one byte each instead of the massive 15 for a POSITION statement.

I am now finding this column hard to compile. Not because I have nothing to write about but mainly because I am limited to one page and the articles I would like to include would take more room. What I need now is readers suggestions, either to me or the Editor, for page size articles. With your help I can explore the areas YOU wish to read about.

You will be reading this in September and, hopefully, I should be in sunny Florida in October. If you want any questions answered directly please write as soon as possible, otherwise you may have to wait many weeks for a reply.

Finally, I hope that by the time you read this I will have my last(?) **FIRST STEPS TUTORIAL** tape out. I hope that they have been of some benefit and, although they were designed for the beginner, I hope that the tapes included something for the more advanced.

Write to Mark at BAUG SOFTWARE, P.O.BOX 10, BELFAST, BT10 0DB

GOTO DIRECTORY

The GOTO DIRECTORY is a guide to retailers who provide product support for Atari. Many of these retailers will supply Mail Order so if you have trouble finding a supplier, turn to the GOTO DIRECTORY.

Retailers who are interested in an entry in this feature are invited to contact the Editor on 0785 41153.

A.S.Wootton & Sons,
116, Edleston Road,
Crewe,
CW2 7HD
Tel: 0270 214118

Nothing but Atari. Authorised Service Centre with fast turnaround of all repairs. The usual range plus printers, joystick inserts, extension leads at very good prices. Try us for repairs or purchases.

Channel One Computer Systems Ltd,
174, High Street,
Hornchurch,
Essex
Tel:Hornchurch (04024) 75613

We have a large range of software plus hardware, printers etc. A good selection of American software including the more unusual such as Compilers, Editors, Assemblers etc. All for Atari.

Home Entertainment Atari Center,
212-213 Broad Street,
Birmingham,
B15 1AY
Tel:021 643 9100

Your dedicated Atari retailer for the best in the wonderful world of Atari. Mail Order Software Courier service - phone 021 643 9100 or write FREEPOST. Atari authorised Independent Service Centre.

Home Entertainment Atari Center,
In-Store Shop,
Lower Ground Floor,
Owen Owen Store,
Mander Centre,
Wolverhampton
Tel: Wolverhampton (0902) 711650

Your in-store Atari Center specialising in Atari related products and including an Atari Service pick up point.

Jennings Stores,
248, Hertford Road,
(Nr. Green Street),
Enfield,
Middx.
Tel. 01 804 1767

Very large range of English and American software available as well as the very latest in hardware.

Microbyte,
71, Seaview Road,
Liscard,
Wallasey,
Merseyside, L45 4QW
Tel. 051 630 6933

Tired of high prices, poor service and hidden charges from other Mail Order retailers? Try us for the latest titles and U.S. mags. Visit or call anytime up to 7pm.

Norman Audio Ltd,
51, Fishergate,
Preston,
Lancs
Tel. 0772 53057

Authorised Atari Service Centre. Dealers in Atari products since 1978. Competitive prices and full after-sales back up. Personal service or Mail Order.

Radford Hi-Fi Ltd,
52, Gloucester Road,
Bristol,
Avon
Tel:0272 428247

We stock and suport a comprehensive range of products for Atari. Huge range of software from educational to small business (plus games of course). Word processing packages. Printers. All for the best computer!

Trionic,
144, Station Road,
Harrow,
Middx HA1 2RH
Tel:01 861 0036

Software, peripherals, books and magazines. A comprehensive range for Atari. Try our late night shopping. Open 10 a.m to 8 p.m. Monday to Saturday. Give us a call or pay us a visit.

York Computer Centre
7, Stonegate Arcade,
York
Tel: 0904 641862

Top American and English software - over 1000 titles! Hardware, books, magazines, accessories. If you need anything for your Atari, try Yorkshire's widest and most comprehensive range of products for your machine.

Programming

RESET ROUTINES

One of the program routines which caused most interest appeared in Issue 6 in the game Scramble. It was a simple means of re-running a program on SYSTEM RESET. The routine shown would not work on the XL models but the following routine will.

```
10 POKE 2,52:POKE 3,185:POKE 9,2:TRAP
 1000
20 GRAPHICS 18:POSITION 1,5:? #6;"pres
s system reset"
30 GOTO 30
1000 RUN
```

A more sophisticated SYSTEM RESET routine is given here. Unfortunately this will not work on the XL models. If anyone can come up with a fix, please let us know.

Cassette users should change lines 30 and 300 as follows.

```
30 I=ADR(R$):H=INT(I/256):L=I-H*256:PO
KE 9,2:POKE 2,L:POKE 3,H:POKE 842,12:G
RAPHICS 0
300 POKE 9,0:POKE 2,0:POKE 3,0:? "NOW
PRESS THE SYSTEM RESET BUTTON":END
```

Keep those prying eyes out of your programs!

```
1 REM ***** SYSTEM RESET ROUTINE *****
5 REM **   USE ESCAPE KEY TO EXIT   **
10 DIM R$(49)
20 R$="))●-E )K (SV) ┐T))♥)R (SV)U (SV)N (SV
)↑ (SV)↑ (SV)▔■J┘L♥█"
30 I=ADR(R$):H=INT(I/256):L=I-H*256:PO
KE 12,L:POKE 13,H:POKE 842,12:GRAPHICS
 0
35 REM ******************************
100 REM ****** DEMONSTRATION ********
105 GRAPHICS 18
110 POKE 16,64:POKE 53774,64:REM DISAB
LE BREAK KEY
115 POSITION 6,4:? #6;"protected"
120 POSITION 6,6:? #6;" program"
130 POSITION 1,8:? #6;"PRESS SYSTEM RE
SET"
140 POKE 710,RND(0)*200
145 IF PEEK(764)=28 THEN GOTO 300:REM
ALLOW ESCAPE THROUGH ESC KEY
150 GOTO 140
200 REM *****************************
299 REM RESET POINTERS TO NORMAL
300 POKE 12,64:POKE 13,21:? "NOW PRESS
 THE SYSTEM RESET BUTTON":END
399 REM
400 REM *****************************
500 REM ROUTINE IN DECIMAL FORM FOR
          INFORMATION
600 DATA 169,148,141.197,2,169,125,32,
164,246,169.2,133,84,169,0,169,82,32,1
64,246,169,85,32,164,246,169,78,32,164
700 DATA 246,169,28,32,164,246,169.28.
32,164,246,169,13,141,74,3,76,0,160
```

More **SCREENDUMPS** from a picture disk in the MACE library. Authors unknown.





From English Software. The supreme Atari challenge. Only £9.95.

Atari 400, 800 & XL COMPATIBLE

For Atari owners, English Software programs are the finest.

Our large and unique range of titles will stretch your imagination to the limit and beyond.

The needle-sharp graphics, vibrant colours and super-smooth action will really test your skills.

English Software is the leading independent for Atari – nobody does it better.

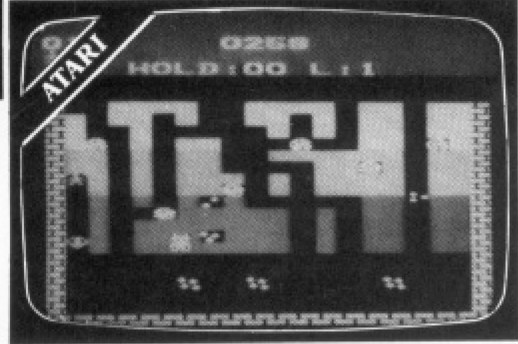
And we're fast making our programs compatible for Commodore 64.

Order direct by using the coupon (or telephone the **Sales Hot Line** on **061-835 1356**, quoting your Access or Visa number).

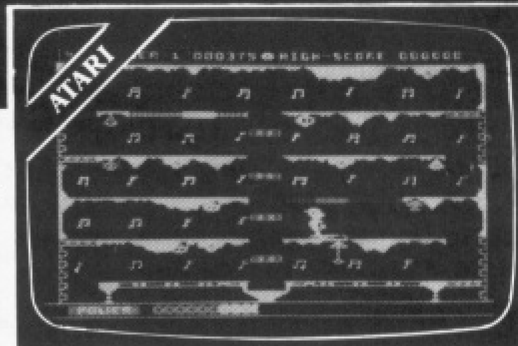


Find us at Boots, Laskys, Greens and all good software dealers.

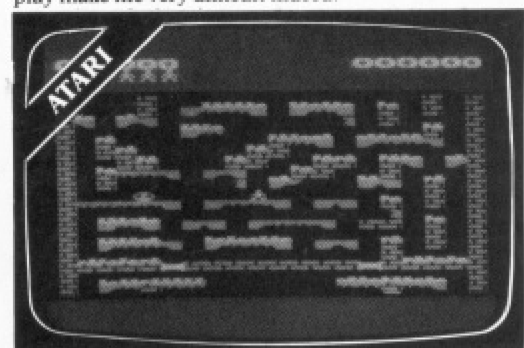
(All English Software programs are sold according to the English Software Terms of Trading, copies of which are available on request.)



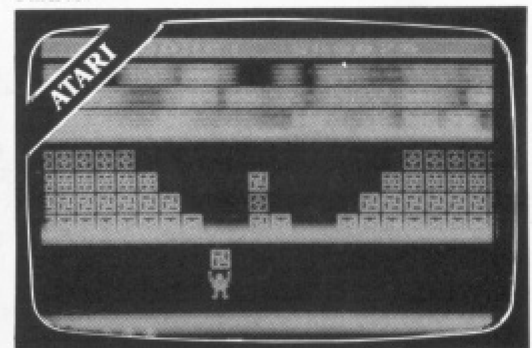
DIAMONDS 16K Cassette/Disk by Simon Hunt. Chase the Great White Diamond in ENGLISH SOFTWARE'S best-selling mining game! 16 levels of play make life very difficult indeed!



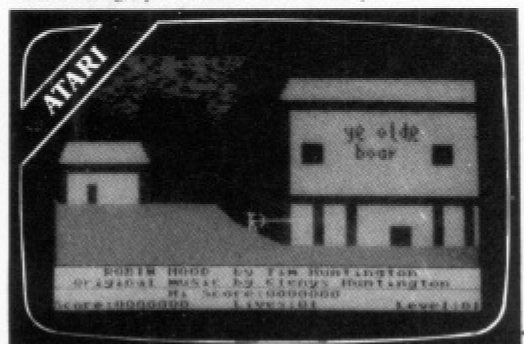
JET-BOOT JACK 32K Cassette/Disk by Jon Williams. JET-BOOT JACK, space-age jogger, takes you on a chase through the vinyl vaults of the PRESSING PLANT!



DAN STRIKES BACK 16K Cassette/Disk by Simon Hunt. In the first sequel to the top-selling DIAMONDS, Brian the Blob has stolen the GREAT DIAMOND and hidden it in the deepest vault.



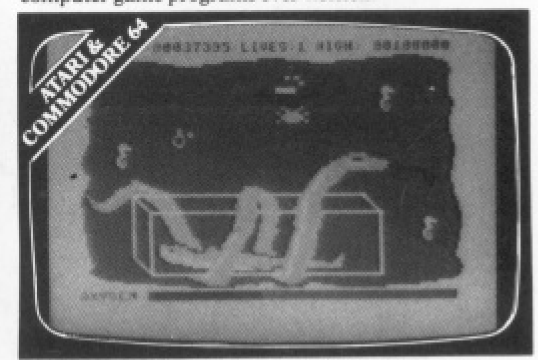
BATTY BUILDERS 16K Cassette/Disk by Manuel D Caballero. Play BATTY BUILDERS and indulge yourself in one of the best non-violent constructive computer game programs ever written.



THE ADVENTURES OF ROBIN HOOD 16K Cassette/Disk by Tim Huntington. Join ROBIN HOOD in his efforts to thwart the SHERIFF'S MEN, rescue the bags of silver and plant kisses on the lips of the beautiful MAID MARIAN!



CITADEL WARRIOR 32K Cassette/Disk by Jon Mayers and Ken Farnen. The nations of the world are at the mercy of D-E-A-T-H (Dastardly Earthwide Anarchists and Terrorists against Happiness.) They are out to destroy the security CITADELS with their CYCLOTRON BOMBS!

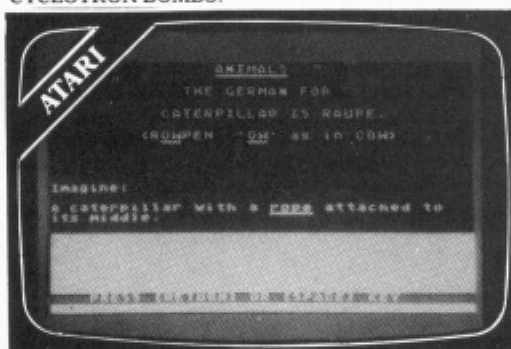


NEPTUNE'S DAUGHTERS 16K Cassette/Disk by Mark J Taylor and Michael Hedley. Designed by Ralph Frumin. Our 1st multi-screen arcade adventure takes you beneath the ocean's waves in search of NEPTUNE'S DAUGHTERS, held captive by the evil Sea Serpent!

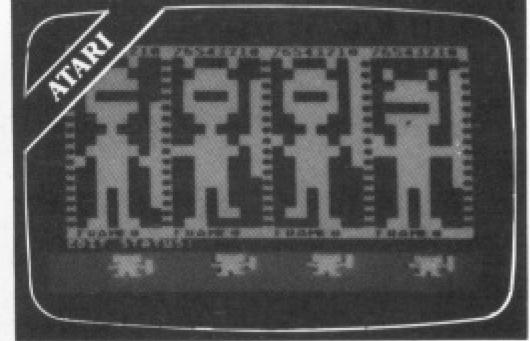
ATARI

A.C.E. THE ATARI CASSETTE ENHANCER by Jon Williams

Cassette features include: Names cassette-saved programs. Searches and loads named program. Displays contents of cassette on screen. Verify facility. Lists variables. Automatic line numbering. Renumbers any basic program quickly. Creates binary files on tape. Plus many, many more features. Uses only 4K of memory maximum! £7.95



GRUNEBERG LINKWORD LANGUAGE SYSTEM: GERMAN 32K Cassette/Disk Designed by Dr Michael M Gruneberg. Program by Steven A Riding. With the use of a unique blend of visual imagery and psychology, it will teach you the basic grammar and more than 350 German words in only 10 hours! Supplied complete with separate audio pronunciation tape. £12.95



THE ATARI GRAPHICS WIZARD! 16K Cassette/32 K Disk by Steven A Riding. THE ATARI GRAPHICS WIZARD introduces you to the wonderful world of ATARI PLAYER (SPRITE) and CHARACTER GRAPHICS! Written totally in Machine Code, it contains: PLAYER EDITOR (SPRITEMAKER); CHARACTER EDITOR, MULTICOLOUR CHARACTER EDITOR.

PROGRAMMERS WANTED

We're always on the lookout for new programmers. If you can work to the English Software standard of quality, we'll reward you by marketing your programs across Europe and USA! Contact us today.



THE POWER OF EXCITEMENT

The English Software Company, Box 43, Manchester M60 3AD Trade Enquiries Tel: 061-835 1358

To English Software Company, Box 43, Manchester M60 3AD.
Please rush me the following on cassette/disk. Tick box
I enclose cheque/PO/Cash for £______ (post-free) or please debit my Access/Visa No ______
Name ______
Address ______

The Adventures of Robin Hood	□	£ 9.95
A.C.E.	□	£ 7.95
Diamonds	□	£ 9.95
Dan Strikes Back	□	£ 9.95
Citadel Warrior	□	£ 9.95
Gruneberg German	□	£12.95
Jet-Boot Jack	□	£ 9.95
Batty Builders	□	£ 9.95
Neptune's Daughters Atari	□	£ 9.95
Neptune's Daughters CBM 64	□	£ 7.95
The Atari Graphics Wizard	□	£ 9.95

Access VISA

P6